Diplomarbeit zum Thema

"Analyse eines neuartigen Horizontalkrankonzepts für Containerterminals mithilfe der Simulationstechnik"

zur Erlangung des akademischen Gerades

Diplom Wirtschaftsinformatiker

vorgelegt am

Department Informatik der Universität Hamburg,

Arbeitsbereich Angewandte und Sozialorientierte Informatik

Nikolaus Alexander Bornhöft

Matrikelnummer: 5428967

Fachsemester 15

7. Dezember 2009

Erstbetreuer: Prof. Dr.-Ing. Bernd Page

Zweitbetreuer: Dr.-Ing. Holger Schütt

Danksagung

Die vorliegende Diplomarbeit geht auf eine Kooperation zwischen dem „Institut für Seeverkehrswirtschaft und Logistik (ISL)“ und dem Arbeitsbereich „Angewandte und sozialorientierte Informatik (ASI)“ des Departments Informatik der Universität Hamburg zurück.

Für die Unterstützung bei meiner Diplomarbeit möchte ich mich bei folgenden Personen herzlich bedanken:

Meinen Betreuern Prof. Dr. Bernd Page und Dr. Holger Schütt für das mir entgegen gebrachte Vertrauen und die Möglichkeit an einem so interessanten und vielseitigen Thema zu arbeiten.

Meiner Mutter Anneliese Bornhöft, meiner Schwester Astrid Lilian Bornhöft, meinem Bruder Hans-Henning Bornhöft und Herrn Holger Brause für ihre Unterstützung und viele Stunden Korrekturlesen.

Herrn Horst-Dieter Kassl für die Hilfestellungen bei der Einarbeitung in Golem 3D und die Containerterminal-Logistik sowie seine ansteckende Begeisterung für die Thematik.

Herrn Johannes Göbel, der mir eine Vielzahl von Fragen zu Simulation, Desmo-J und LaTeX mit großer Expertise und viel Geduld beantworten konnte.

Meinem Kommilitonen Felix Klückmann, mit dem ich die letzten Monate den Diplomantenraum geteilt habe und der mir zu dieser Zeit Stütze und Motivation war.

Inhaltsverzeichnis

1. Einleitung

Hafenlogistik

Der weltweite Warenhandel hat in den vergangenen 50 Jahren rasant zugenommen. Mit einem durchschnittlichen Wachstum von 6,1% lag er weit oberhalb der Steigerung der Weltwarenproduktion von 3,9% [Bil]. Innerhalb des Welthandels verzeichnet der Containerhandel die größten Zuwächse. Auf den Containerhäfen der Nordrange - Hamburg, Bremische Häfen[1] , Rotterdam und Antwerpen hat sich der Containerumschlag in den Jahren 1991 bis 2007 von 10 Mio TEU[2] auf knapp 33 Mio TEU mehr als verdreifacht ([Fle08], S.10). Gleichzeitig nimmt die Zahl großer Containerschiffe mit einer Kapazität von mehr als 10000 TEU erheblich zu. Waren Anfang 2008 erst 9 Containerschiffe dieser Größe im Einsatz, werden es Anfang 2012, den aktuellen Fertigungsaufträgen nach, bereits 173 ([Fle08], S.12) sein. Daraus ergibt sich für die Betreiber von Containerterminals noch stärker als bisher die Notwendigkeit effizienter Be- und Entladevorgänge. Sie ergibt sich zum einen, weil der Anteil des Containerumschlags gemessen an der Gesamtliegezeit bei größeren Schiffen zunimmt und zum anderen weil die Kapitalkosten für die Reeder bei größeren Schiffen höher sind und eine starke Konkurrenzsituation zwischen den einzelnen Containerterminals, insbesondere denen der Nordrange, herrscht. Mit der Zunahme des weltweiten Containerumschlags ergibt sich die Notwendigkeit der Erweiterung und des Baus neuer Terminals. Dabei besteht die Möglichkeit mit neuartigen Containerterminal-Architekturen Effizienzgewinne und damit Wettbewerbsvorteile zu erhalten.

[1] Bremische Häfen fasst die Häfen in Bremen und Bremerhaven zusammen.
[2] Twenty-Feet-Equivalent Unit, Standardmaß für Container

Das Horizontalkrankonzept der Firma ZPMC

Das in dieser Arbeit betrachtete Horizontalkrankonzept ist eine neuartige Architektur für den Containertransport auf einem Terminal. Es wurde unter dem Namen „Environmentally Friendly Automated Container Yard Stacking System“ von der Firma „Shanghai Zhenhua Heavy Industry Co., Ltd.“, ZPMC, einer Tochter der „China Communication Construction Co., Ltd.“ (CCCC) [ZPMa] entwickelt. Das in Shanghai beheimatete Unternehmen ist der weltweit größte Hersteller von Schwerlastanlagen. Die Hauptgeschäftsbereiche bilden dabei Krane für Containerterminals, insbesondere Kaikrane, hochbelastbare Stahlbauten z.B. für Brücken oder Kraftwerke und Schwerlast-Offshore-Anlagentechnik für Bohrplattformen und Windparks.

Der Bau oder die Erweiterung eines Containerterminals sind mit hohen Kosten und langen Bauzeiten verbunden. Daher muss die gewählte Architektur schon vor Baubeginn auf ihrer Eignung überprüft werden, einen effizienten Betrieb im späteren Containerterminal zu ermöglichen. Bei Architekturen, die schon in anderen Terminals Anwendung finden, kann dabei auf Erfahrungswerte zurückgegriffen werden. Da es sich hier jedoch um ein neues Konzept handelt und dessen Erfinder, ZPMC, als Hersteller von Kranen und Schwerlastanlagen ein starkes Interesse daran hat, Containerterminal-Architekturen mit einem hohen Anteil an Krananlagen zu verkaufen, muss sich der Betreiber des Terminals ein eigenes, unabhängiges Bild von der Leistungsfähigkeit des Konzepts machen. Dazu können zum einen analytische Methoden und zum anderen Simulation dienen.

Mathematische Methoden der Analytik eignen sich dabei nur für einige Teilbereiche der Untersuchung von Containerterminal-Prozessen. Probleme, die viele variable Ein- und Ausgangsgrößen besitzen, verfügen oft über eine Komplexität, die rechnerisch nicht lösbar ist. Mit den Mitteln der Simulation können sie jedoch abgeschätzt werden.

Zielsetzung und Herangehensweise

Ziel der vorliegenden Diplomarbeit ist die Untersuchung des Horizontalkrankonzeptes der Firma ZPMC hinsichtlich seiner Anwendbarkeit in einem Containerterminal. Dazu wird das Modell eines Beispielterminals entworfen, in dem das Konzept verwendet wird.

Zunächst werden in Kapitel 2 zentrale Begriffe geklärt und die Betriebsmittel und Architekturen bestehender Containerterterminals vorgestellt. Ihnen wird in einem weiteren Schritt die Idee des Horizontalkrankonzepts gegenübergestellt. Dabei wird den neuartigen Komponenten und ihren Interaktionen sowie den Wechselwirkungen mit bestehenden Containerterminal-Komponenten besondere Beachtung geschenkt.

Bei der Untersuchung soll die diskrete Ereignissimulation Anwendung finden. Darüber hinaus sollen die Simulationsläufe in Form einer Animation abspielbar sein. Dafür werden in Kapitel 3 unterschiedliche Programme auf ihre Eignung überprüft, das Horizontalkrankonzept in einem Simulationsmodell abzubilden, daran Experimente durchzuführen, diese statistisch auszuwerten und die Ergebnisse zu animieren. Anschließend wird eine Auswahl getroffen und diese begründet.

Zur Erstellung des Simualtionsmodells wird in Kapitel 4 zunächst ein Containerterminal des neuartigen Typs entworfen, das in seiner Größe und dem Umfang des Containerumschlags mit bereits bestehenden Containerterminals vergleichbar ist. Dadurch sollen auch die dort gewonnenen Kennzahlen eine Vergleichbarkeit mit denen realer Containerterminals erhalten und zu bewertenden Aussagen über das Horizontalkrankonzept führen.

Dazu müssen vor der Erstellung des Computermodells die modellrelevanten Systemkomponenten mit ihren spezifischen Eigenschaften und ihren Positionen identifiziert werden. Des Weiteren werden Abstraktionen von nicht simulationsrelevantem Verhalten und Vereinfachungen zur Abbildung der Steuerung vorgenommen.

In Kapitel 5 wird die Erstellung des Simulationsmodells beschrieben. Dabei werden alle zuvor identifizierten Systemkomponenten in das Modell abgebildet. Die Steuerung wird gemäß der in Kapitel 4 definierten Abstraktionen und Vereinfachungen zu einem Teil direkt in das Verhalten einzelnen Modellkomponenten und zu einem anderen Teil in einer globalen Steuerung abgebildet.

Kapitel 6 beschäftigt sich mit den Modellein- und ausgaben. Für die Eingabe der Modellparameter wird eine grafische Benutzerschnittstelle angelegt, in der ein Szenario mit spezifischen Werten für den Simulationslauf, die Dimensionierung des Modells und Schiffs- und Auftragsankünften beschrieben werden kann. Die Ausgabe untergliedert sich in die Erzeugung einer Animationsdatei und einer Excel-Datei, die statistische Kennzahlen zum Simulationsexperiment enthält.

Über mehrere Simulationsexperimente werden in Kapitel 7 Kennzahlen für das Containerterminal-Modell gewonnen. Diese orientieren sich vergleichend an einem bestehenden Containerterminal, so dass durch sie eine qualifizierende Bewertung vorgenommen werden kann.

Im Fazit in Kapitel 8 wird die vorgenommene Modellierung retrospektiv betrachtet. Es werden Stärken und Schwächen des Modells diskutiert und mögliche Erweiterungen und Änderungen aufgezeigt. Weiterhin wird eine differenzierte, abschließende Beurteilung des Horizontalkrankonzeptes abgegeben.

2. Horizontalkrankonzept

Zur Einordnung des in dieser Arbeit vorgestellten Horizontalkrankonzeptes soll zunächst auf Terminalarchitekturen eingegangen werden, die bereits im laufenden Betrieb Anwendung finden und anhand derer die Grundstruktur eines Containerterminals erkennbar wird. Dann soll die Architektur des Horizontalkrankonzepts mit ihren spezifischen Komponenten vorgestellt und erläutert werden.

2.1. Bestehende Containerterminal-Architekturen

Im Grundaufbau befindet sich bei allen Containerterminals seeseitig die „Berthing-Area“, ein Bereich, der Liegeplätze für Containerschiffe und Containerbrücken zum Be- und Entladen umfasst. Gates sind Verladestationen. Sie sind landseitig die Schnittstelle zum Bahn- und LKW-Transport. Zentral befindet sich außerdem die „Stacking-Area“, in der Container bis zu ihrem Weitertransport gelagert werden. Sie lässt sich weiter in gewöhnliche Stellflächen, Stellflächen für Tiefkühlcontainer, die jeweils einen Stromanschluss besitzen, sowie Flächen für Leercontainer unterteilen. Der Bereich, auf dem in dieser Arbeit das Hauptaugenmerk liegt, befindet sich zwischen der Berthing- und der Stacking-Area. Die dortigen Anlagen und Fahrzeuge transportieren die Container zwischen den Kaikranen und dem Containereinstellplatz im Lagerblock (Abb. 2.1).

Generell lassen sich verschiedene Architekturen von Containerterminals anhand ihres Automatisierungsgrades und der eingesetzten Krane und Transportmaschinen unterscheiden. An dieser Stelle werden zunächst jene Komponenten vorgestellt, die in den meisten Terminals weltweit in gleicher oder ähnlicher Form Anwendung finden. Dann werden exemplarisch zwei bestehende Architekturvarianten des terminalinternen Transportes unterschieden.

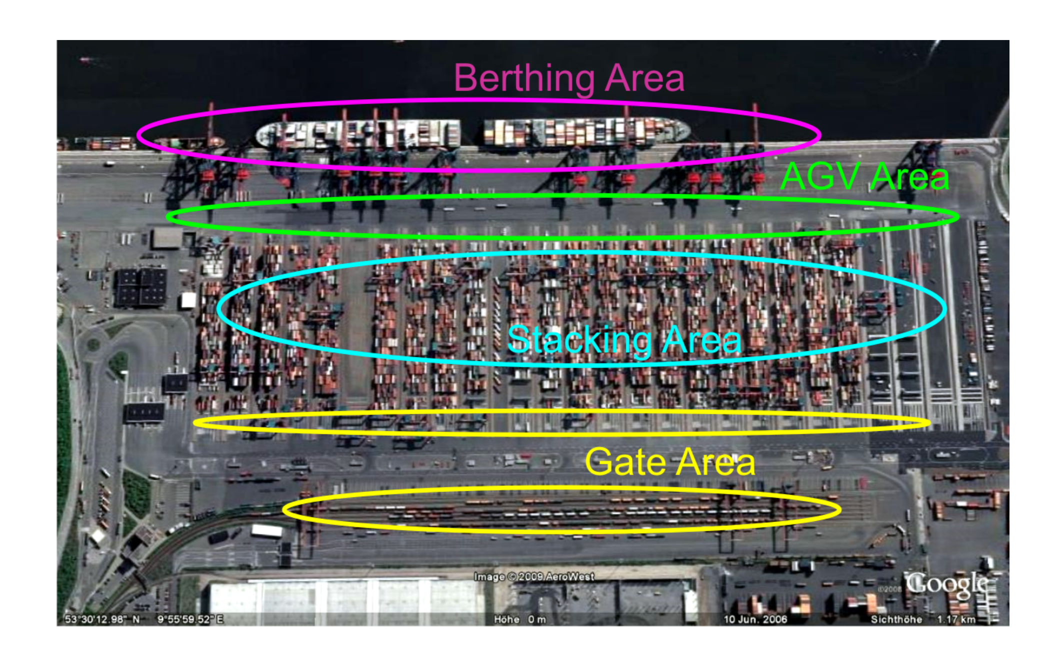


Abb. 2.1.: **Aufbau eines Containerterminals am Beispiel des CT-Altenwerder (modifiziert aus [Goo])**

2.1.1. Die Berthing-Area

Die Berthing-Area ist der wasserseitig abgeschlossene Umschlagbereich eines Containerterminals. Dort werden entlang der Kaimauer ankommende Containerschiffe zum Be- und Entladen festgemacht.

Containerschiffe

Containerschiffe sind ein Typ Frachter, der auf den Transport von Containern ausgelegt ist (Abb. 2.2). Den weitaus größten Anteil am weltweiten Containerhandel haben dabei Vollcontainerschiffe, die ausschließlich zum Containertransport verwendet werden ([Sch89], S.13). Auf ihnen soll hier das Hauptaugenmerk liegen.

Containerschiffe werden in der Regel in Längsstau beladen. Das bedeutet, dass die Container auf dem Schiff längs zur Fahrtrichtung stehen. Diese Vorgehensweise wird deswegen verwandt, weil die physikalischen Kräfte, die durch Seegang auf die Ladung wirken in Schiffsquerrichtung stärker als in Längsschiffrichtung sind und die Containerstirnwände weniger stark sind als die Seitenwände ([GDV], Platzierung und Sicherung der Container



Abb. 2.2.: **Containerfrachter [Nau]**

an Bord). Die Ausrichtung des Spreaders[1] der Containerbrücke ist dementsprechend so, dass die Container in Schiffslängsrichtung geladen und gelöscht werden.

Die Staubereiche auf einem Containerschiff werden in verschiedene Bereiche unterteilt. Zum Verständnis der Terminologie werden diese hier benannt und in einer hierarchischen Ordnung vorgestellt. Moderne Containerschiffe werden sowohl über als auch unter Deck beladen. Dabei werden die Container unter Deck in Staugerüste[2] gestapelt, die horizontale Bewegungen der Container verhindern und weitere Transportsicherungen überflüssig machen. Auf einer Grundfläche können mehrere Container übereinander gestapelt werden. Ein solcher Stapel wird als Stack[3] bezeichnet, Container gleicher Stapelhöhe als Lage[4]. Über den Stack und die Lage lässt sich jede Containerposition also exakt bestimmen (Abb. 2.3).

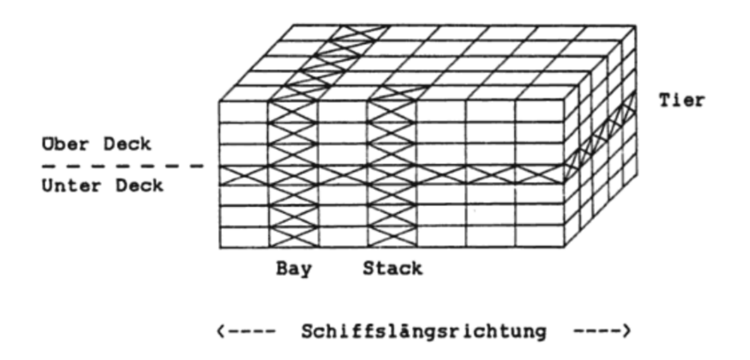


Abb. 2.3.: **Abstrahiertes Schiffstellplatzsystem [Sch89], S.16**

[1]Spreader, engl.: Heberahmen zum Verbinden eines Containers
[2]vertikale Führungsschinen zur Fixierung der Container
[3]Stapel: engl. Stack
[4]engl. Tier

Benachbarte Stacks über die gesamte Breite des Schiffes bilden eine Bay[5]. Je zwei benachbarte Bays werden zu einer Doppelbay zusammengefasst. Maximal zwei benachbarte Doppelbays bilden zusammen einen Laderaum, der durch feuer- und wasserundurchlässige Wände abgeteilt ist. Wände in Schiffslängsrichtung unterteilen die Doppelbays wiederum in je zwei bis drei Luken[6]. Ist unter Deck eine Luke gefüllt, wird sie mit einem Lukendeckel[7] verschlossen, der ähnlich einem gewöhnlichen Container von einer Containerbrücke bewegt wird. Der Lukendeckel schützt die Container in der Luke vor eindringendem Wasser und bildet gleichzeitig den Untergrund für an Deck gestapelte Container. Container innerhalb einer Luke können also nur gelöscht werden, wenn vorher an Deck alle auf der Luke stehenden Container entfernt wurden. ([Sch89], S.13-20 [Gie04])

Containerschiffe lassen sich generell bezüglich ihrer Größe und der Entfernungen für die sie genutzt werden unterteilen.



Abb. 2.4.: **Feederschiff [Bet]**

Relativ kleine Containerfrachter, die als lokale Zubringer eines Containerterminals dienen, werden dabei als Feederschiff[8] bezeichnet (Abb. 2.4). Sie können wegen ihres geringeren Tiefgangs auch kleinere Küsten- und Binnenhäfen ansteuern. Die größten auf dem Rhein fahrenden Binnenfeeder besitzen aktuell eine Kapazität von bis zu 500 TEU, die im Ostseeraum eingesetzten etwa 1400 TEU [Wikb].

[5]Bay, engl.: Containerbucht
[6]Luke: engl. Hatch
[7]Lukendeckel: engl. Hatch Cover
[8]Feeder, engl.: Zubringer

Die größeren Containerschiffe fahren weltweit Häfen an. Seit Beginn der Containerschifffahrt wurden stetig größere Schiffe mit höherer Kapazität entwickelt. Sie stellen damit auch zunehmend höhere Anforderungen an die Wassertiefe der Containerhäfen und Wasserwege.

Das aktuell größte Containerschiff ist die Emma-Maersk mit einer geschätzten Kapazität von 14.500 TEU [Wika].

Containerbrücken

An Land stehen den Containerschiffen zum Be- und Entladen Containerbrücken gegenüber (Abb. 2.5). Diese großen Flurförderzeuge sind schienengebunden und können so entlang der Kailinie parallel zum zu löschenden Containerschiff bewegt werden.

Die Träger einer Containerbrücke überspannen seeseitig ein vor Anker liegendes Containerschiff und landseitig einen Abstell- oder Übergabebereich für Container. Auf dem Brückenträger befindet sich der Fahrerstand. Zudem ist dort noch eine Laufkatze angebracht, die mit einem Spreader - einem Heberahmen für den Container - diese anheben, absenken und zwischen Schiff und landseitigem Containerübergabebereich transportieren kann. [Sch08]

Die Größe einer Containerbrücke wird davon bestimmt, für welchen Einsatz sie angeschafft wurde. Sie muss dabei eine ausreichende Höhe haben, um alle von ihr zu bearbeitenden Containerschiffe so zu überspannen, dass oberhalb eines voll beladenen Schiffes noch genügend Raum zum Bewegen der Laufkatze mit einem Container am Spreader ist. Des weiteren muss der Ausleger des Krans so dimensioniert sein, dass das breitest mögliche zu be- oder entladene Containerschiff komplett im Arbeitsbereich der Laufkatze liegt, so dass jeder Containerstack mit dem Spreader erreichbar ist. Die größten aktuell eingesetzten Containerbrücken erreichen dabei eine Höhe von 80 m und sind gut 1800 t schwer [Abe].

Eine Architekturvariante stellen Zwei-Katz-Containerbrücken dar. Bei diesen ist der Containertransport in zwei Phasen unterteilt. Eine so genannte Laschplattform bildet dabei einen Puffer zwischen den Arbeitsbereichen der beiden Laufkatzen. Da jede Katze

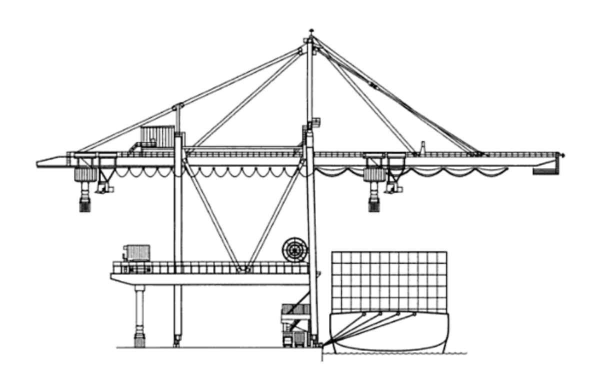


Abb. 2.5.: **Zwei-Katz-Containerbrücke mit Laschplattform, [Sch08], S.206**

nur etwa die Hälfte des gesamten von der Brücke zu transportierenden Weges bearbeitet, können insgesamt Produktivitätsgewinne erreicht werden [HS00].

Die Laschplattform ist gleichzeitig der Bereich, in dem Arbeiter die für den Seetransport nötigen Transportsicherungen von den Containern entfernen.

2.1.2. Die Stacking-Area

Die Stacking-Area bildet den internen Lagerbereich eines Containerterminals. Dort werden ankommende Container bis zu ihrer Weiterverschiffung oder ihrem Weitertransport per Schiene oder Strasse zwischengelagert. So können Verspätungen ausgeglichen, die LKW- und Bahnankünfte an den Gates zeitlich besser verteilt und bestimmte Arbeitszeitvorgaben eingehalten werden.

Die Art, wie die Container zwischengelagert werden, unterscheidet sich zum einen zwischen unterschiedlichen Containerterminals mit unterschiedlichen Betriebsmitteln und zum anderen innerhalb eines Terminals bezüglich der verschiedenen Containertypen.

Die unterschiedlichen Betriebsmittel bedingen, welche Abstände zwischen den einzustellenden Containern eingehalten werden müssen und in welcher Weise die einzelnen Containerblöcke angeordnet werden (Abschnitt 2.1.3).



Abb. 2.6.: **Portalkran auf einem Containerblock [Inf]**

Stacking-Area Bereiche nach unterschiedlichen Containertypen

Bei den verschiedenen Containertypen gibt es größenbedingte Restriktionen bezüglich ihrer Stapelbarkeit. Neben den 20 ft und 40 ft Containern gibt es noch einige andere, weltweit verschiffte, Containerformate, insbesondere den 45 ft High-Cube Container. Diese sind unter Umständen nicht untereinander stapelbar.

Neben Containern verschiedener Größen gibt es noch weitere Typisierungen, anhand derer Container in besonderen Bereichen untergebracht werden müssen. So benötigen einige Container einen eigenen Stromanschluss zur Kühlung bzw. Temperierung der Güter. Weiterhin werden Leercontainer gesondert und zumeist höher gestapelt. Bei Schwerlastcontainern muss darauf geachtet werden, dass sie nicht so hoch gestapelt werden dürfen wie Standardcontainer. Gefahrgutcontainer bedürfen ebenfalls einer besonderen Lagerung [Con].

Containerblöcke mit eigenem Betriebsmittel

Bei Containerterminals, in denen der terminalinterne Transport mit Betriebsmitteln mit eigener Hubfähigkeit wie zum Beispiel Straddle-Carriern (Abb. 2.7) durchgeführt wird, müssen Containerstellplätze inklusive ihrer Stapelhöhe und der Bereiche zum Rangieren und Arbeiten, auf die technischen Merkmale der Fahrzeuge abgestimmt, definiert werden.

Wird der terminalinterne Containertransport mittels Fahrzeugen ohne eigene Hubfähigkeit, also auf einer Art Chassis durchgeführt, bedarf es eines Krans, der das Chassis be- und entlädt und den transportierten Container an passender Stelle in den Lagerblock hebt oder ihn dort entnimmt. Typischerweise werden dazu Portalkrane[9] verwendet, die rad- oder schienengestützt auf genau einem Lagerblock arbeiten.

Der Portalkran wird so bezeichnet, weil er in Form eines großen Stahlrahmens seinen Arbeitsbereich wie ein Portal überspannt (Abb. 2.6). Auf dem oberen Querträger des Rahmens befinden sich der Kranführerstand und eine Laufkatze, die über die Breite des Portals bewegt werden kann. Über einen an der Katze hängenden Spreader werden, analog zum Spreader an einer Containerbrücke, Container aus dem Block ein- oder ausgelagert. Bei einem Portalkran sind neben den Geschwindigkeiten von Laufkatze und Spreader noch weitere technische Merkmale unterscheidbar. Schienengeführte Portalkrane sind, im Gegensatz zu radgeführten, in der Regel auch mit einem Container am Spreader beweglich. Des weiteren gibt es Portalkrane, die sich, gemeinsam auf einem Containerblock arbeitend, gegenseitig über- und unterfahren können.

Im Gegensatz zur Organisation der Stacking-Area für Straddle-Carrier oder Reach-Stacker, bei der zwischen den einzelnen Stellplätzen und Stellplatzreihen Platz für Fahrspuren und zum Rangieren der Fahrzeuge gelassen werden muss, ist die Stacking-Area hier in kompakte Blöcke unterteilt. Zwischen den Stellplätzen wird dabei gerade so viel Raum gelassen, wie der Spreader des Portalkrans zum Zugriff auf einen einzelnen Containerslot benötigt. Damit benötigen Containerterminals mit Blöcken und Portalkranen insgesamt weniger Platz für die Stacking-Area als solche mit keiner kompakten Blocklagerung.

Ein Containerblock lässt sich als mehrere benachbarte Containerstellplatzreihen auffassen, für die es eine gemeinsame Höchststapelhöhe gibt. Da der Block von einem Portalkran überspannt wird, bildet die Spannweite des Portalkrans eine technische Grenze für die maximale Breite eines Blocks. Ebenso ist die maximale Blockhöhe nicht nur von den physikalischen Stapeleigenschaften der Container sondern auch von den technischen Merkmalen des Krans abhängig. Innerhalb dieser technisch-physikalischen Grenzen spielen Optimierungsgesichtspunkte eine entscheidende Rolle. Zwar kann ein Containerblock

[9]Portalkran: engl. Gantry Crane

theoretisch in beliebiger Länge angelegt werden, jedoch korreliert die Länge der mittleren zurückzulegenden Wege der Portalkrane direkt mit der Blocklänge. Des weiteren steigt bei kurzen Containerblöcken auch mit guten Prestacking-Strategien die Anzahl der nötigen Umstapelaufträge, um Container zu erreichen, die nicht zu oberst auf einem Stack liegen.

Die Anzahl und die technische Ausgestaltung der auf einem Containerblock arbeitenden Portalkrane hängt neben der Länge, Höhe und Kapazität der Blöcke auch davon ab, wie diese mit den anderen Betriebsmitteln des Terminals interagieren. Werden die Container an der Stirnseite des Stacks an den Portalkran übergeben, so dass dieser den Transportweg zum Containerslot fahren muss, so müssen weitere Wege zurückgelegt werden, als wenn der Container von einem Chassis oder AGV parallel zum Stack transportiert und erst auf Höhe des Containerslots übergeben wird. Ebenso wird die Ressource Portalkran stärker beansprucht, wenn dieser für weitere Aufgaben z.B. das Beladen von LKW an den Gates eingesetzt wird.

2.1.3. Bereich des terminalinternen Containertransports

Weit verbreitet sind Terminals, auf denen bemannte Fahrzeuge wie Gabelstapler oder Van-Carrier die Container zwischen der Berthing-Area und den Lagerblöcken transportieren. Diese Fahrzeuge verfügen über die Fähigkeit, Container eigenständig anzuheben und auf mehrfache Containerhöhe zu stapeln. So bedarf es bei dieser Architektur keines weiteren Krans zum Ein- und Auslagern im Stack. Nachteilig wirkt sich hier aus, dass innerhalb der Stacking-Area vergleichsweise viel Raum zwischen den Containerstellplätzen für Fahrspuren der eingesetzten Fahrzeuge benötigt wird. Van-Carrier kommen unter anderem im Terminal „Burchardkai“ [HHLc] der Hamburger Hafen und Logistik AG, HHLA, sowie dem „Northsea Terminal“ Bremerhaven zum Einsatz [NTB].

In den derzeit am stärksten automatisierten Containerterminals weltweit werden „Automatisch Geführte Fahrzeuge“ (AGV[10]) (Abb. 2.8) zum Transport der Container zwischen den Lagerblöcken und den Kaikranen eingesetzt [Gota]. Diese sind unbemannte Fahrplattformen, die ausschließlich passiv be- und entladen werden können und die selbständig, einem Verkehrsleitsystem folgend, Transportaufträge durchführen. Da sie über

[10]engl. Automated Guided Vehicles: automatisch geführte Fahrzeuge



Abb. 2.7.: **Straddle Carrier [Kon]**

keine Hubfähigkeit verfügen, werden neben den Containerbrücken weitere Krane zum ein- und auslagern in die Lagerblöcke benötigt. Eingesetzt wird diese Architektur im ECT Delta Terminal in Rotterdam [ECT] und dem Containerterminal Altenwerder der HHLA [HHLb], dem derzeit am weitesten automatisierten Terminal weltweit [Sch04].



Abb. 2.8.: **Automated Guided Vehicle [HHLa]**

2.2. Architektur der Horizontalkrankonzepts

Der Einsatz der Horizontalkrane stellt eine Designalternative zu bestehenden Containerterminals dar. Dabei wird der Transport der Container zwischen den Kaikranen und

den Lagerblöcken, anders als bei aktuell in Betrieb befindlichen Containerterminals in zwei Schritte unterteilt. Es gibt einen Transportschritt parallel zur Kaimauer über eine auf etwa 7 m hohen Stelzen konstruierte Schienenverbindung, die Low-Frame-Bridge (Abb. 2.9). In einen Schritt werden die Container mittels eines unbemannten Transportchassis zwischen der Low-Frame-Bridge und einem Portalkran am Einstellplatz des Containers im Stack transportiert.

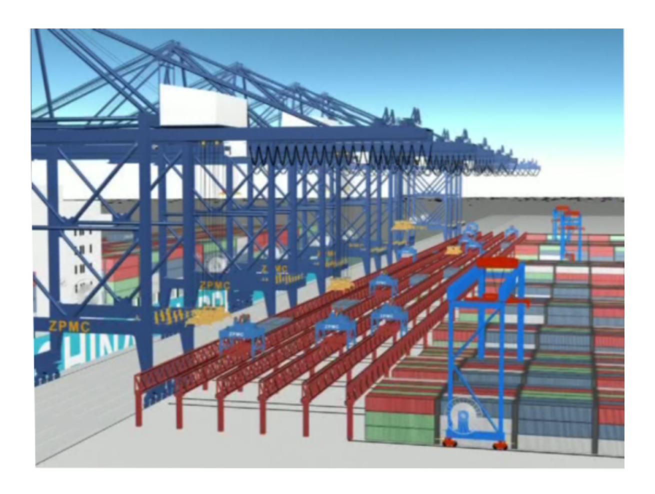


Abb. 2.9.: **Low-Frame-Bridge**

Die Low-Frame-Bridge bildet das innovative Herzstück des Horizontalkrankonzeptes. Sie ist eine ein- oder mehrspurige Schienenkonstruktion, die, getragen von Stützpfeilern, unterhalb der Containerbrücke verläuft. Auf ihr arbeiten Flat-Trolleys und Lifting-Trolleys.

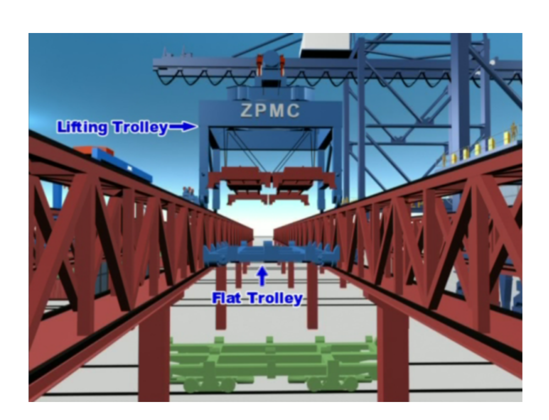


Abb. 2.10.: **Flat-Trolley und Lifting-Trolley**

Flat- und Lifting-Trolley Flat-Trolleys sind Schlitten, mit denen auf einer Spur Container befördert werden. Sie können direkt von den Containerbrücken be- oder entladen werden. Ebenso wie die Flat-Trolleys bewegen sich auch die Lifting-Trolleys schienengeführt auf einer Horizontalspur der Low-Frame-Bridge (Abb. 2.10). Beide Gerätetypen können sich dabei gegenseitig über- bzw. unterfahren. Lifting-Trolleys bilden die Schnittstelle zwischen den Horizontalspuren und darunter operierenden Transportchassis. Befindet sich ein beladener Flat-Trolley auf Höhe des Stacks in den der Container eingelagert werden soll, fährt ein Lifting-Trolley an dieselbe Position und hebt den Container an. Daraufhin verlässt der Flat-Trolley den Bereich. Der Lifting-Trolley senkt den Container, dreht ihn und setzt ihn auf einem wartenden Transportchassis ab (Abb. 2.11).

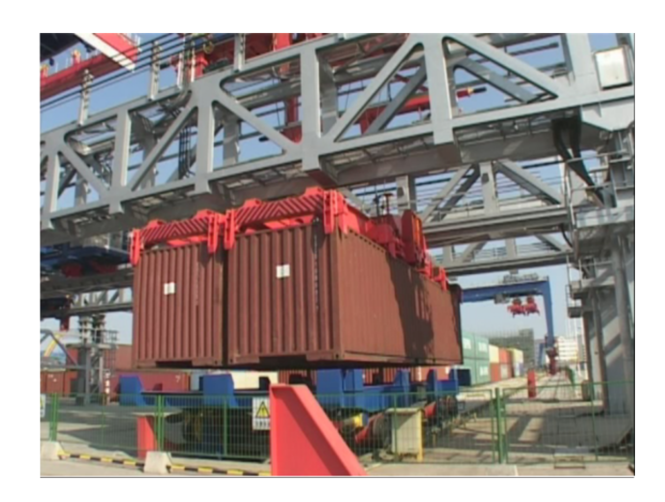


Abb. 2.11.: **Ein Lifting-Trolley setzt Container auf einem Chassis ab**

Das Chassis führt den nächsten Transportschritt hin zu der Stelle im Lagerblock durch, an der der Container eingestellt werden soll und wird dort von einem Portalkran entladen. Anders als bei einem AGV ist das Chassis hier in seiner Bewegung auf einen festen Bereich entlang des Lagerblocks, für den es zuständig ist, festgelegt.

Der Einsatz des Horizontalkrankonzeptes bedeutet gegenüber einer AGV-Architektur eine erhebliche Vereinfachung der Steuerungslogik in den einzelnen Systemkomponenten, da immer auf eine exklusive Nutzung bestimmter Bereiche oder Ressourcen abgezielt wird. So arbeitet jedes Transportchassis an genau einem Lagerblock und nutzt seine Transportspur allein. Flat-Trolleys können bei mehreren Horizontalspuren auf der Low-

Frame-Bridge zwar parallel eingesetzt werden, auf einer Horizontalspur findet jedoch wieder ein exklusiver Zugriff auf den zu befahrenden Bereich der Spur statt. Neben der Vereinfachung der Steuerung in den Einzelkomponenten bringt das Horizontalkrankonzept einen Mehraufwand bei der Synchronisation der Teilkomponenten mit sich, da bei einem Transportauftrag an der Low-Frame-Bridge je ein Flat-Trolley, ein Lifting-Trolley und ein Transportchassis an einer Containerübergabe beteiligt sind. Eine weitere Besonderheit des Horizontalkrankonzeptes ist, dass sich jede Transportkomponente ausschließlich über eine gerade Strecke bewegt. Damit sind alle Containerübergabepunkte zwischen zwei Komponenten im Vorhinein bekannt. Eine passende Steuerung kann die Synchronisation zwischen zwei Betriebsmitteln dementsprechend besser planen, da sich aus Position und Geschwindigkeit der Komponenten das jeweilige Eintreffen am Synchronisationspunkt bestimmen lässt.

Sind die beteiligten Komponenten eines Containertransports ausreichend aufeinander abgestimmt und die Teilsysteme derart dimensioniert, dass keine der beteiligten Komponenten einen Flaschenhals bildet, könnten bei ihrem Einsatz Effizienzgewinne gegenüber klassischen Containerterminals erreicht werden. Der jeweilige Anteil einer Komponente am Gesamttransport eines Containers könnte vermindert und somit mehr Container gleichzeitig transportiert werden.

3. Simulation

In diesem Kapitel sollen zunächst die Begriffe Simulation im Allgemeinen und diskrete Ereignissimulation im Speziellen erklärt werden. Dann wird die Auswahl der Simulations- und Animationswerkzeuge für die vorliegende Modellierung diskutiert und zuletzt das gewählte Simulationswerkzeug, Desmo-J, vorgestellt.

Simulation bezeichnet allgemein Systemanalyse mittels Durchführung von Experimenten am Modell eines Systems anstelle des Systems selbst (Abb. 3.1). Um dabei Rückschlüsse auf das Originalsystem ziehen zu können, muss die Abbildung zwischen Realsystem und Modell hinreichend korrekt sein ([Pag91], S. 7). Die Art, was für ein Modell erstellt wird, wie es erstellt wird und wie Beobachtungen daran vorgenommen werden, kann mitunter stark variieren. Man unterscheidet zunächst statische und dynamische Modelle. Ein statisches Modell beschreibt ein System in einem Zustand, ein dynamisches die Veränderungen der Modellzustände über die Zeit. Dynamische Modelle lassen sich weiter in diskrete und kontinuierliche Modelle untergliedern. Kontinuierliche Modelle werden durch kontinuierliche Zustandsänderungen des Modells über Zeiträume beschrieben. Typische Anwendungsbereiche kontinuierlicher Simulation bilden Untersuchungen physikalischer Prozesse. Sie sind mathematisch oft komplex und beinhalten die Verwendung von Differentialgleichungen zur Beschreibung der Zustandsänderungen im System über die Zeit.

Diskrete Modelle werden dort eingesetzt, wo eine Abstraktion der Zustandsänderungen des Systems auf Zeitpunkte möglich ist und man jede Änderung des Modellzustands bereits zu einem vorherigen Zeitpunkt unter Angabe der Art und des Zeitpunkts der Zustandsänderung spezifizieren kann. Sowohl diskrete als auch kontinuierliche Simulationsmodelle lassen sich in deterministische und stochastische unterteilen. Deterministische Modelle lassen sich analytisch beschreiben. Sie zeigen für dieselben Eingangswerte immer

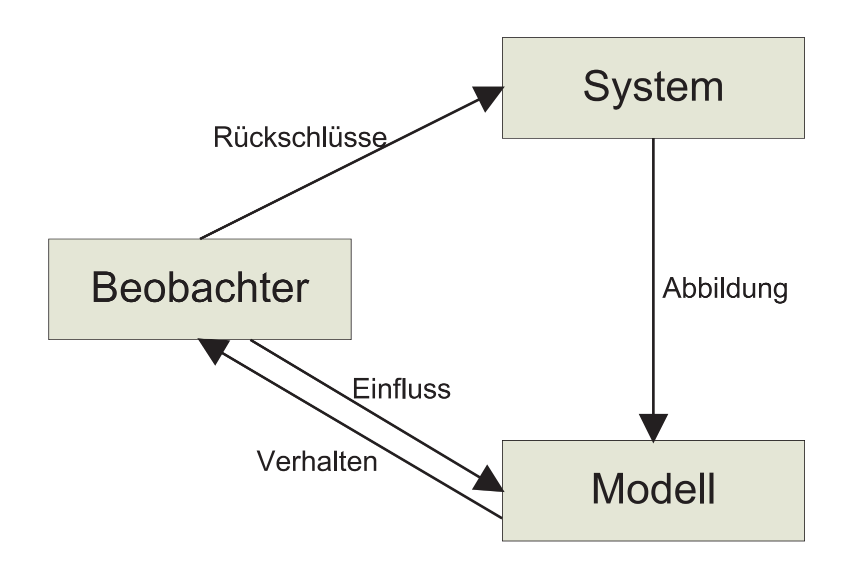


Abb. 3.1.: **Verhältnis von System, Modell und Beobachter, verändert aus [PK05], S.9**

dasselbe Modellverhalten. Stochastische Modelle enthalten an mindestens einer Stelle die Abbildung zufälligen Verhaltens. Dieses wird in der Computersimulation durch die Verwendung von Werten einer Wahrscheinlichkeitsverteilung realisiert. Da das Modell unter denselben Bedingungen unterschiedliches Verhalten zeigt, müssen für eine valide Beschreibung des Verhaltens mehrere Simulationsläufe statistisch ausgewertet werden. Die Verwendung stochastischer Modelle bietet die Möglichkeit über das Verhalten von Systemen Aussagen zu treffen, die rein analytisch nicht oder nur schwer zu beschreiben sind.([PK05], S. 7-12)

3.1. Wahl des Simulationswerkzeuges

Zur Untersuchung des Horizontalkrankonzepts wird das Modell eines Containerterminals entworfen, in dem die Komponenten des Konzeptes Anwendung finden. Es soll dabei in seinen Dimensionen einem bestehenden Terminal nachempfunden werden, so dass Experimente am Modell zu Werten führen, die Rückschlüsse auf das Konzept zulassen. Weiterhin soll eine Animation der Ergebnisse ermöglicht werden, die das Verständnis der Interaktionen der einzelnen Komponenten erleichtert.

Dazu werden zunächst verschiedene Werkzeuge zur Modellierung und Animation diskreter Simulation vergleichend untersucht.

Nach einer ersten Vorauswahl aus verschiedenen Programmen erscheint Desmo-J [des], ein Java-Simulationsframework, das am Department Informatik der Universität Hamburg entstand, geeignet um das Simulationsmodell zu erstellen. Für die Animation erweist sich der Realtime Animation Player Golem 3D [ISLa] des ISL als passend. Alternativ besteht die Überlegung die Simulationssoftware Plant Simulation [Sied] der Firma Siemens zu verwenden. Diese ermöglicht eine grafische Modellerstellung und eine Animation direkt aus dem Modell heraus. Zu untersuchen ist, falls Plant Simulation für die Modellierung des Modells verwendet wird, ob die integrierte Animation verwendet werden oder ob zusätzlich eine Schnittstelle zu Golem 3D erstellt werden soll. Um eine qualifizierende Bewertung für die Auswahl zu erhalten, werden die Möglichkeiten der Modellerstellung in Desmo-J und Plant Simulation, sowie die der Animation in Plant Simulation und Golem 3D einander gegenüber gestellt.

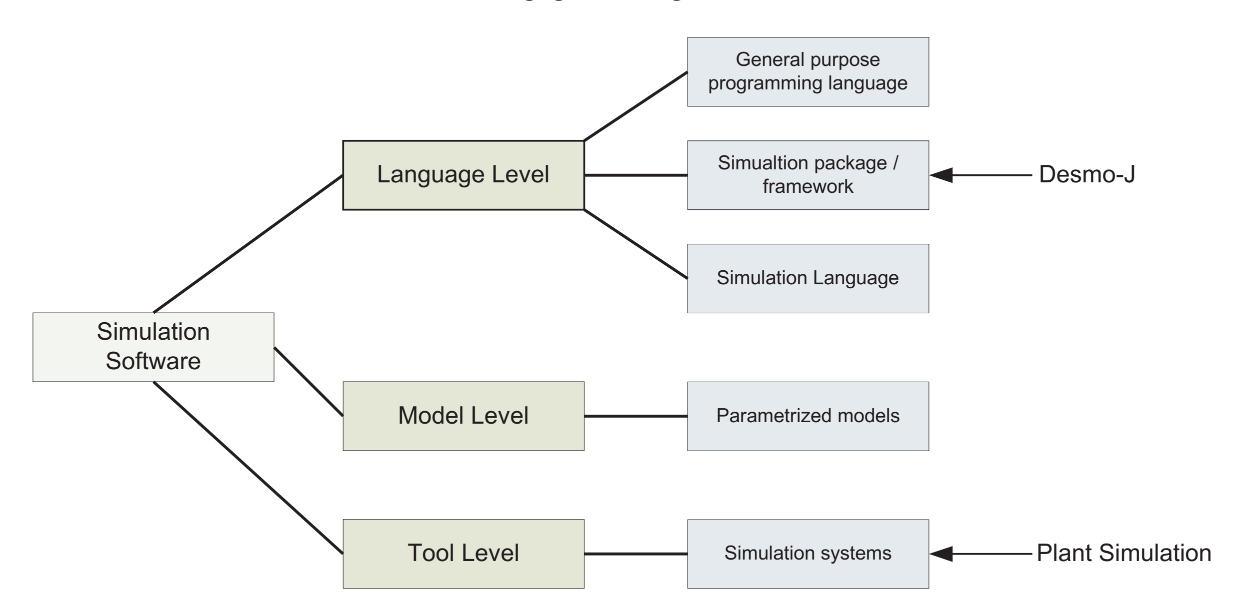


Abb. 3.2.: **Klassifikation von Plant Simulation und Desmo-J, verändert aus [PK05]**

Vergleich Plant Simulation und Desmo-J

Desmo-J ist ein Java-Simulationsframework. Es bietet eine Infrastruktur für die Erstellung von Simulationsmodellen und die Durchführung von Experimenten. Dazu liefert es

dem Anwender zum einen Black-Box-Komponenten, die als Simulationsumgebung verwendet werden und nicht weiter verändert werden müssen, und zum anderen Whitebox-Komponenten, die dem anwendungsspezifischen Verhalten der Modell-Komponenten entsprechend angepasst erstellt werden. Die Black-Box-Komponenten umfassen den Scheduler, die Abbildung von Wahrscheinlichkeitsverteilungen und Warteschlangen. Die White-Box-Komponenten sind Simulationsereignisse und -prozesse. Sie besitzen Methoden zur Beschreibung des Kontrollflusses im Modell über einen standardisierten Zugriff auf den Scheduler und sind bezüglich ihres Verhaltens auf Programmcode-Ebene vollständig zu spezifizieren.

Plant Simulation ist ein Simulationssystem, das eine Vielzahl fertiger Modell-Komponenten bereithält, die grafisch per Drag and Drop auf einer Oberfläche, dem Netzwerk, angeordnet werden. Der Materialfluss des Modells, zum Beispiel die Bewegung von Containern, zwischen den Komponenten lässt sich ebenfalls grafisch über Verbindungen realisieren (Abb. 3.3). Das modellspezifische Verhalten wird in Teilen an den Komponenten parametrisiert. Dabei ist insbesondere jenes Verhalten gut über Standardparameter abbildbar, das sich in Form von Bedienstationen, die von Werkstücken durchlaufen werden, verstehen lässt. Das Warten auf die Verfügbarkeit von Komponenten wird über das „Schiebe- und Blockier“-Prinzip realisiert, das alle Materialfluss-Objekte, die an einer Komponente ankommen, automatisch in einer Warteschlange zurückhält bis die Komponente verfügbar ist. Komplexe Steuerungslogik wird in Plant Simulation mit „uneigentlichen Objekten“ realisiert, die als logische Modell-Entitäten vorliegen. Modellverhalten innerhalb der „uneigentlichen Objekte“ wird in der prozeduralen Sprache „SimTalk“ spezifiziert. Sim Talk bietet zahlreiche Möglichkeiten statistischer Auswertung, der Erstellung und Animation von 3D-Modellen und der Zufallszahlengenerierung. [Tec]

Für die Verwendung von Desmo-J spricht, da es auf der Programmiersprache Java basiert, die Verfügbarkeit besserer Entwicklungsumgebungen zur Programmierung und die stärkere Strukturierung der Sprache gegenüber der prozeduralen Sprache SimTalk, die als proprietäre Sprache nicht so umfassend dokumentiert ist, wie es bei Java der Fall ist. Des weiteren liegt Desmo-J unter der Apache License [ASF04] vor und ist damit frei benutzbar. Da das Framework am Arbeitsbereich ASI entwickelt wurde, ist die inhaltliche Unterstützung bei der Betreuung von Seiten der Universität für dieses Softwarewerkzeug besser. Frameworkfremde Programmteile wie eine über den Standard hinausgehen-

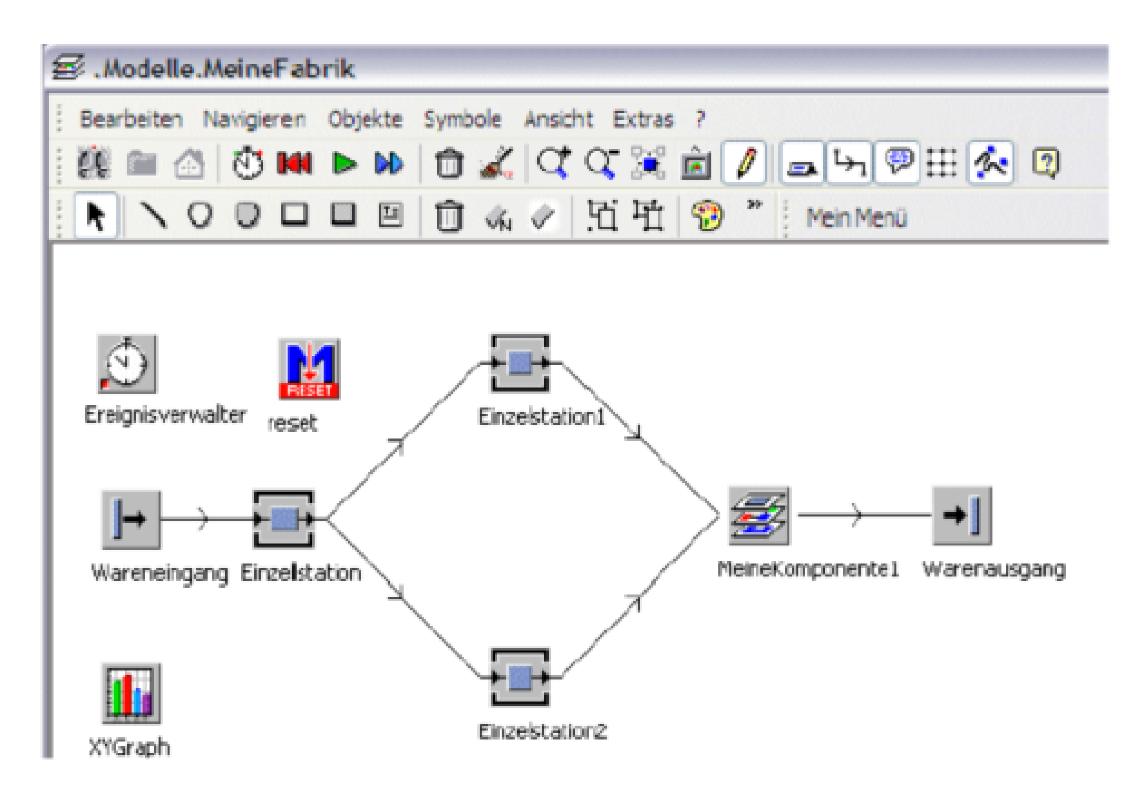


Abb. 3.3.: **Beispielmodellierung in Plant Simulation**

de Auswertungskomponente und eine grafische Benutzeroberfläche können nahtlos in die Programmstruktur des Simulationsmodells eingefügt werden, da die Sprache Java zu Grunde liegt und es für diese eine Vielzahl von Erweiterungsbibliotheken gibt.

Für die Verwendung von Plant Simulation spricht, dass es als umfangreiches Simulationssystem die Modellierung auf einem höheren Abstraktionsniveau erlaubt und eine Vielzahl fertiger Komponenten schon im Voraus besitzt, die nur noch angepasst und verwendet werden müssen. Das der grafischen Modellierung zu Grunde liegende Raumkonzept wird implizit übernommen und es sind keine Brüche zwischen einzelnen Programmen zur Übergabe von zum Beispiel Modelldateien mehr nötig. [BK08]

Vergleich der Erstellung einer Animation in Plant Simulation und Golem 3D

Sowohl Plant Simulation als auch Golem 3D erlauben die Animation eines Simulationsmodells in einem dreidimensionalen Raum. Die Animation in Golem 3D kann bei der

vorliegenden Modellierung durch die Wiedergabe einer während der Simulation erstellten Animationsdatei realisiert werden. Die Animation in Plant Simulation erfolgt direkt durch die Simulation in einem 3D-Simulationsmodells.

Die Vorteile einer Nutzung von Plant Simulation bestehen darin, dass bei der direkten Modellierung des 3D-Simulationsmodells die Brüche zwischen den einzelnen Programmen sowie das Einlesen und die Ausgabe von Modell- und Animationsdateien wegfallen.

Für die Nutzung von Golem 3D spricht neben der fachlichen Expertise, auf die bei der Betreuung durch das ISL zurückgegriffen werden kann, dass dort zahlreiche Komponenten für die Animation von Containerterminals schon vorhanden sind. Es müssen nur noch die für das Horizontalkrankonzept spezifischen Betriebsmittel Flat- und Lifting-Trolley sowie die Low-Frame-Bridge grafisch modelliert werden. Als Eigenentwicklung des ISL ist Golem 3D dort zur Nutzung bereits vorhanden.

Die ausgewählten Werkzeuge

Die Wahl der Simulations- und Animationswerkzeuge für die vorliegende Modellierung fällt auf Desmo-J und Golem 3D. Neben den Kenntnissen auf Seiten der Betreuer dieser Arbeit, auf die zurückgegriffen werden kann, scheinen sich die Werkzeuge gut zu ergänzen. Sie decken gemeinsam die vollen Anforderungen ab, die durch das Thema der Arbeit gestellt sind. Das Modell-Layout, das im 2D-Editor Verdigo für eine spätere Nutzung in Golem 3D erstellt wird, kann mithilfe von Java-Bibliotheken zur Verarbeitung von XML-Dokumenten eingelesen werden und stellt so das in Desmo-J nicht spezifizierte Raumkonzept zur Verfügung. Desmo-J stellt als Framework sämtliches Verhalten, das für die diskrete Ereignissimulation benötigt wird, zur Verfügung und bietet darüber hinaus die gesamte Ausdrucksmächtigkeit der Sprache Java zur Abbildung des modellspezifischen Verhaltens.

3.2. Desmo-J

Die Abkürzung Desmo-J steht für „Discrete Event Simulation and MOdelling in Java[1]“. Desmo-J wurde am Arbeitsbereich ASI des Departments Informatik der Universität Hamburg in der Arbeitsgruppe von Professor Page im Rahmen studentischer Projekte entwickelt, 1999 in seinen Kernfunktionalitäten fertiggestellt und seitdem kontinuierlich weiterentwickelt. Das Framework geht auf das 1989 in Modula 2 geschriebene Original-DESMO zurück, das seinerseits Anleihen an dem 1979 in der Simulationssprache Simula entstandenen Paket DEMOS für diskrete Ereignissimulation nimmt. ([des], Quick Overview)

Black-Box- und White-Box-Komponenten

Desmo-J ist ein Framework zur objektorientierten Softwareentwicklung für diskrete Ereignissimulation. Es unterstützt den Entwickler bei der Erstellung von Simulationsmodellen, der Durchführung von Experimenten an diesen und deren Auswertungen. Dazu stellt es so genannte „Black-Box-Komponenten“ und „White-Box-Komponenten“ zur Verfügung. Die Black-Box-Komponenten bilden Klassen, von denen der Entwickler nur ein Objekt erzeugen muss, dieses parametrisieren kann und es, wie es ist, verwenden kann. Sie bilden die Komponenten ab, die ein rein simulationsspezifisches Verhalten beschreiben und für alle Modelle in gleicher Weise eingesetzt werden. Die Black-Box-Komponenten umfassen die unterschiedlichen Warteschlangentypen, stochastische Verteilungen, statistische Auswertungskomponenten und die Experimentklasse. In White-Box-Komponenten wird das anwendungsspezifische Verhalten abgebildet. Dazu werden dem Anwender abstrakte Klassen zur Verfügung gestellt, bei deren Implementierung er die Anwendungsspezifika in seinem Modell realisiert. Sie definieren das aktive Verhalten der Modellkomponenten während der Simulation als Entität, Simulationsprozess, Ereignis oder externes Ereignis (Abb. 3.4). ([PK05], S.263-267)

[1] engl.; diskrete Ereignis-Simulation und- Modellierung in Java

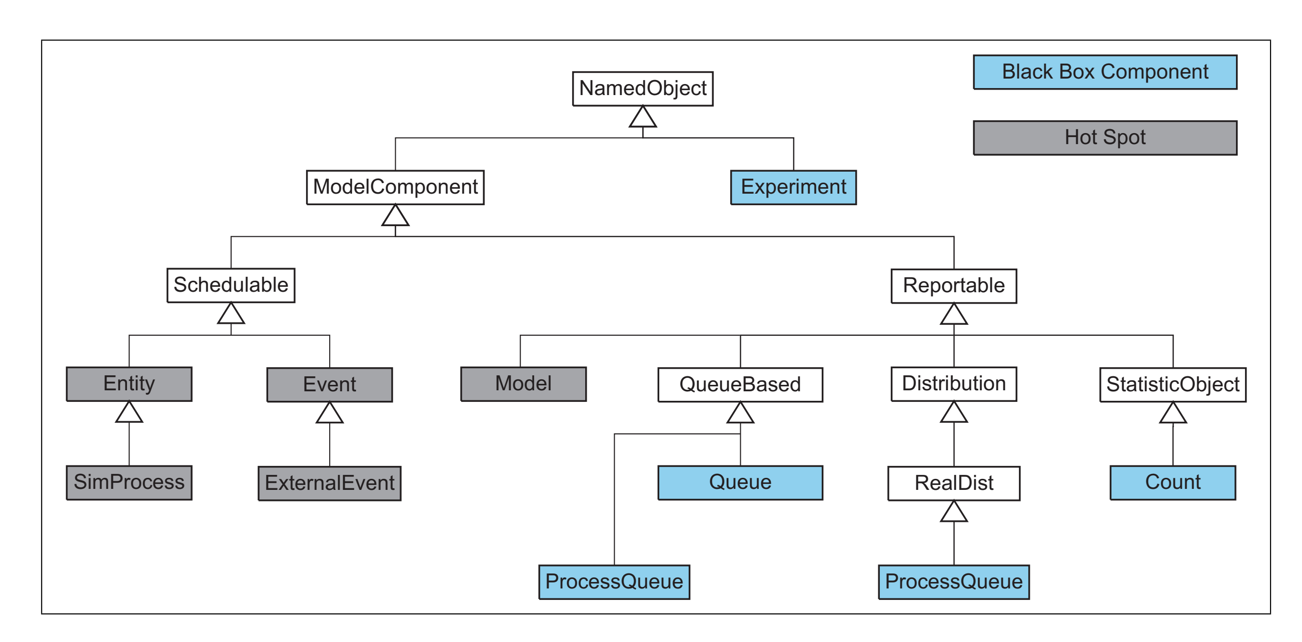


Abb. 3.4.: **Aufbau der Klassenstruktur von Desmo-J, [PK05], S.266**

Scheduler und Ereignisliste

In der diskreten Ereignissimulation passieren alle Zustandsänderungen eines Modells zu bestimmten Zeitpunkten. Von kontinuierlichen Änderungen wird abstrahiert.

Ein Simulationsmodell enthält alle modellspezifischen Entitäten und die Art und Weise, wie und wann Ereignisse stattfinden und was sie bewirken in statischer Form. Zur Durchführung eines Simulationslaufs wird das Modell an ein Experiment gebunden. Das Experiment kapselt intern die beiden zentralen Elemente zur Abbildung des Zeitfortschritts, die Ereignisliste und den Scheduler. Die Ereignisliste enthält Ereignisse, mit einer festen Reihenfolge und einem jeweiligen Zeitpunkt, in dem sich stattfinden. Der Scheduler löst Ereignisse zu ihrem Ausführungszeitpunkt aus. Sind alle Ereignisse und das durch sie bedingte Verhalten in einem Zeitpunkt durchgeführt, erhöht der Scheduler die Modellzeit auf den Zeitpunkt des nächsten Ereignisses aus der Ereignisliste und aktiviert es. Ereignisse können bei ihrer Ausführung den Modellzustand und die Ereignisliste verändern. Liegen nachdem die Kontrolle an den Scheduler zurückgegeben wurde keine Elemente mehr in der Ereignisliste vor, oder ist der definierte End-Zeitpunkt des Experiments erreicht, beendet der Scheduler den Simulationslauf (Abb. 3.5).

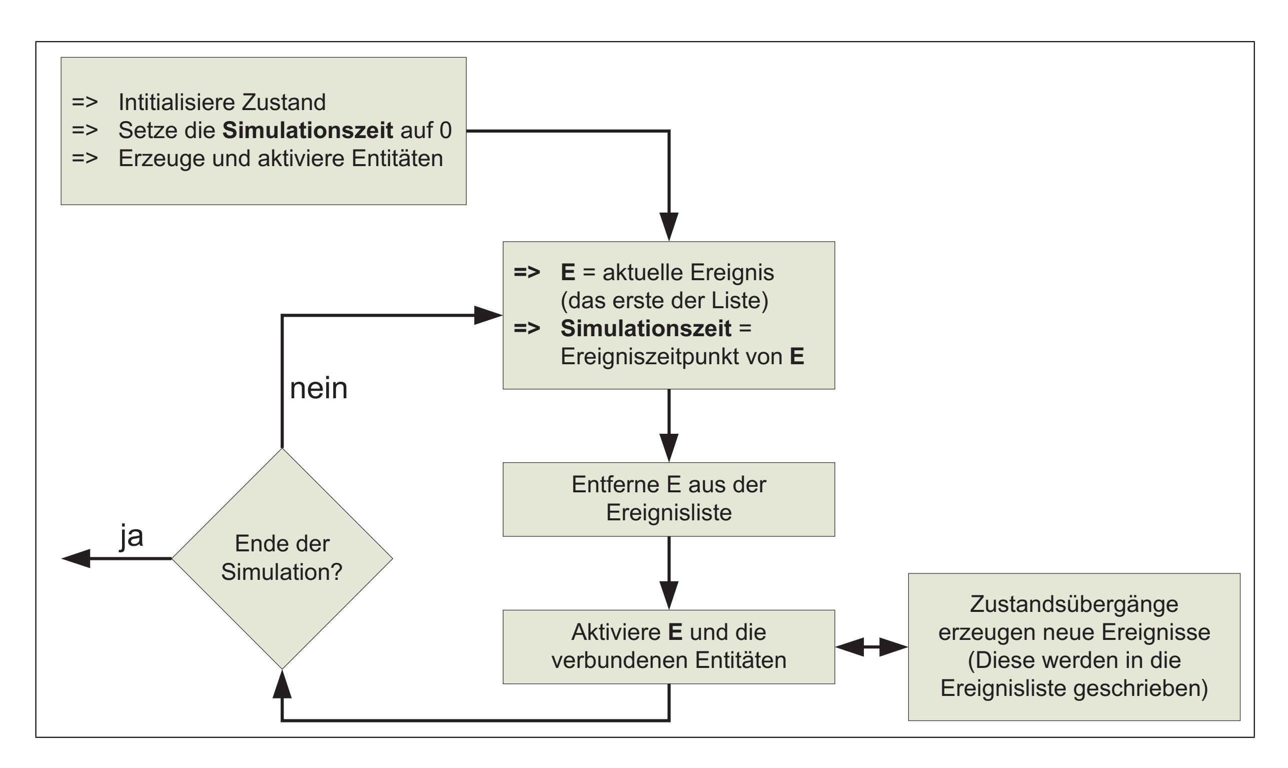


Abb. 3.5.: **Realisierung des Zeitfortschritts im Scheduler, verändert aus [PK05], S.26**

Ereignisorientierter Modellierungsstil

Desmo-J ermöglicht verschiedene Modellierungsstile, nämlich einen ereignisorientierten und einen prozessorientierten sowie hybride Mischformen.

Die ereignisorientierte Modellierung entspricht dabei am stärksten der internen Repräsentation der Ereignisse in Ereignisliste und Scheduler. Die verwendeten Elemente der ereignisorientierten Modellierung sind Objekte der Typen `Entity`[2] und `Event`[3] sowie `SimTime`. Die Entität beschreibt dabei den Zustand einer Modellkomponente. Ein Ereignis spezifiziert eine Zustandsänderung im Modell. Jedes Ereignis wird dabei genau einer Entität zugeordnet, auf die es sich bezieht. Neben der Veränderung der Attribute von Entitäten können durch ein Ereignis weitere Ereignisse in die Ereignisliste des Schedulers eingefügt werden und dort befindliche entfernt werden. Das Einfügen kann zu einem spezifizierten Zeitpunkt oder zum selben Zeitpunkt eines im Scheduler befindlichen

[2]engl.; Entität
[3]engl.; Ereignis

Ereignisses logisch vor oder nach diesem vorgenommen werden. Das `SimTime`-Objekt repräsentiert entweder eine Zeitdauer, wenn es als Parameter dem Scheduler übergeben wird,

```
schedule(Event what, SimTime when)
```

oder einen Zeitpunkt, wenn es den Rückgabewert einer Methode spezifiziert.

Der ereignisorientierte Modellierungsstil entspricht der Sichtweise auf das Modell aus der Vogelperspektive, weil sich Ereignisse zwar aus eine Entität beziehen, jedoch nicht direkt in der Entität erzeugt werden, sondern auf einer höheren Ebene definiert werden.

Prozessorientierter Modellierungsstil

Der prozessorientierte Modellierungsstil nimmt statt der Vogelperspektive des ereignisorientierten Stils die Froschperspektive ein. Simulationsprozesse bilden das Verhalten einer Entität über die Zeit ab. Sie interagieren mit anderen Prozessen und realisieren zeitkonsumierende Tätigkeiten. Sie haben dabei einen veränderlichen Zustand bezüglich des Modells und einen Zustand bezüglich des Kontrollflusses im Programm, der jeweils entweder aktiv oder passiv sein kann.

Aktiv bezüglich des Kontrollflusses bedeutet, dass gerade der Programmcode des Prozesses ausgeführt wird, aktiv bezüglich des Simulationsmodells, dass die abgebildete Modellkomponente gerade einer zeitkonsumierenden Tätigkeit nachgeht und die Kontrolle nach Ablauf der Dauer der Tätigkeit zurückerhält. Ein Simulationsprozess kann immer nur entweder aus Programmsicht oder aus Modellsicht aktiv sein, da der aus Programmsicht aktive Prozess die Kontrolle abgibt, sobald er die Tätigkeit beginnt. Umgekehrt kann keine Modellzeit vergehen, solange die Kontrolle nicht beim Scheduler ist. Prozesse können auch sowohl aus Programmsicht als auch aus Modellsicht passiv sein. ([PK05], S.273)

In einem `SimProcess` werden Übergänge zwischen diesen Zuständen in der Methode `lifeCycle()` spezifiziert. (Abb. 3.6) Der Process kann dabei mit `hold(SimTime dt)` eine Tätigkeit aus Modellsicht beginnen, nach der er die Kontrolle zurück erlangt, mit `activate()` andere passive Prozesse aktivieren und mit `passivate()` selbst die Kontrolle abgeben.

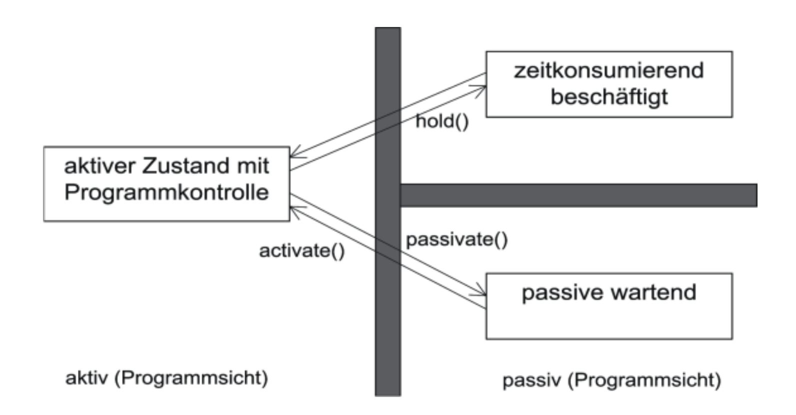


Abb. 3.6.: **Zustandsübergänge von `SimProcess` [PK05], S.273**

Hybrider Modellierungsstil

Desmo-J ermöglicht es den ereignis- und den prozessorientierten Simulationsstil zu koppeln.

Für den Scheduler besteht intern kein Unterschied zwischen der Repräsentation von Ereignissen und dem Verhalten von Simulationsprozessen. Bei Simulationsprozessen wird das Aktivieren eines Prozesses zu einem bestimmten Zeitpunkt als Ereignis vorgemerkt. Ebenso wird für eine zeitkonsumierende Tätigkeit mit `hold(SimTime time)` ein Aktivierungszeitpunkt in der Ereignisliste vorgemerkt. Die beim ereignisorientierten Modellierungsstil im Ereignis spezifizierte Steuerungslogik macht in einem `SimProcess` den Bereich seines `lifeCycle()`s aus, der vom Punkt seiner Aktivierung abgearbeitet wird, bis er die Programmkontrolle wieder abgibt.

Die Verbindung von Prozess- und Ereignisorientierung wird derart umgesetzt, das Prozesse in ihrem Lebenszyklus Ereignisse die Ereignisliste des Schedulers einfügen und von dort entfernen können. Ereignisse können auf Prozesse einwirken, indem sie sie aktivieren oder sie während ihrer konzeptuell beschäftigten „hold“-Phase unterbrechen.

4. Entwurf des Beispielterminals

Zur Untersuchung des Horizontalkrankonzeptes muss zunächst ein Terminal entworfen werden, welches eine Umgebung für die neuartigen Krankomponenten bildet. Innerhalb dieser Umgebung können die Wechselwirkungen mit den übrigen Terminalkomponenten beobachtet und über die Simulation des Gesamtterminals Produktivitätsaussagen getroffen werden. Dazu werden zunächst bestehende Containerterminals betrachtet und passende Systemkomponenten ausgewählt, die später im Modellterminal den Rahmen für die Komponenten des Horizontalkrankonzeptes bilden. Des weiteren wird die Dimensionierung des Beipielterminals vorgenommen, das bedeutet, die Anzahl der benötigten Containerstell- und Schiffsliegeplätze wird bestimmt und auf einen konkreten Terminal-Aufbau abgebildet. Es werden die Interaktionen der Betriebsmittel untersucht und aus ihnen eine explizite Steuerung für das Modell generiert. Die im Rahmen der Modellierung getroffenen Abstraktionen implizieren einige Annahmen für die nicht explizit abgebildeten Teile des Systems, insbesondere die Stau- und die Stellplatzplanung. Diese Annahmen sollen formuliert werden, da sie später helfen die Simulationsergebnisse zu bewerten. In einem letzten Schritt werden die 3D-Objekte der Modellkomponenten in einem 3D-Designer erstellt und die Komponenten in einem Modell-Layout angeordnet, das ein Raumkonzept für die Simulation bietet und gleichzeitig die Grundlage für eine spätere Animation liefert.

4.1. Systemanalyse

Zur Erstellung eines aussagekräftigen Simulationsmodells müssen jene Betriebsmittel eines Containerterminals mit Horizontalkrankonzept identifiziert werden, die für die Bildung eines Modells wesentlich sind ([Pag91], S.10ff). Wesentlich bedeutet in diesem Fall,

dass sie erheblichen Einfluss auf das abzubildende Systemverhalten haben und die Simulationsergebnisse bezüglich der gestellten Aufgabe beeinflussen. Die Designentscheidungen, welche Systemkomponenten in welchem Detaillierungsgrad abgebildet werden, werden im Folgenden genannt und erklärt.

4.1.1. Identifikation der zu modellierenden Betriebsmittel

Die Kernkomponenten des Horizontalkrankonzeptes sind die Low-Frame-Bridge, auf der Flat- und Lifting-Trolleys arbeiten sowie Terminal-Chassis, die Container zwischen der Low-Frame-Bridge und den Portalkranen im Stack transportieren. Als dynamisch interagierende Systemkomponeten sind davon die Trolleys und die Terminal-Chassis abzubilden, die Low-Frame-Bridge wird der Anschaulichkeit halber jedoch in der Animation sichtbar sein. Die übrigen dynamischen Modellkomponenten sind die Containerbrücken und die Portalkrane, welche Container in einem Lagerblock ein- und auslagern. Daneben gibt es LKW und Containerschiffe, die für den Containertransport im System Außengrenzen bilden und Lagerblöcke, die einen Containerslot als Start- bzw. Endpunkt für jeden Transportauftrag bereitstellen.

4.1.2. Gewählte Abstraktionen

Entscheidend bei der Modellierung eines Systems ist die Wahl eines angemessenen Abstraktionsgrades ([Pag91], S.10f). Dabei müssen bezüglich der zu Grunde liegenden Fragestellung nicht nur die zu modellierenden Elemente des Originals, also des Ausgangssystems, ausgemacht werden (Abschnitt 4.1.1), sondern auch die relevanten Merkmale derselben.

Bei der Untersuchung des Horizontalkrankonzeptes liegt das Hauptaugenmerk auf der Interaktion der unterschiedlichen Betriebsmittel. Damit sind der Transport von Containern durch das System sowie der zurückzulegende Weg und die benötigten Geräte die Beobachtungsobjekte, nicht jedoch der Container selbst entscheidend. Deswegen werden die Stell- und Staupläne der Schiffe und Containerblöcke nicht explizit modelliert. Diese werden jedoch in soweit antizipiert, dass generell von einem Vorstau ausgegangen wird,

so dass sich Container für das Beladen eines Schiffes ausschließlich in der seeseitigen Hälfte der Containerblöcke befinden.

Des weiteren wird darauf verzichtet unterschiedliche Containertypen und -arten zu unterscheiden. Unter der Prämisse, dass die vorliegenden Betriebsmittel entweder einen 40 ft Container oder zwei 20 ft Container zur Zeit transportieren können und die 20 ft Container ähnliche Start- und Zielpunkte haben, werden jeweils zwei 20 ft Container oder ein 40ft Container auf einen Transportauftrag abgebildet. Die unterschiedlichen in einem Containerterminal umgeschlagenen Containertypen sind unter anderem Kühl- und Leercontainer. Kühlcontainer benötigen einen Containerstellplatz mit Energieversorgung, ebenso werden Leercontainer gesondert gelagert. Eine Modellierung dieser unterschiedlichen Typen wird nicht vorgenommen, es kann jedoch eine Charge von Transportaufträgen auf einen einzigen bestimmten Lagerblock festgelegt werden.

Bei der Be- und Entladung von Schiffen in einem realen Containerterminal werden die Containerbrücken entlang des Schiffs bewegt, um mehrere benachbarte Schotts zu bearbeiten. Da bei der vorliegenden Modellierung auf eine explizite Abbildung der Stellplätze und Staupläne verzichtet wird, wird auch vom expliziten Zugriff auf einzelne Containerslots abstrahiert. Die Transportaufträge an jeder Containerbrücke beginnen bzw. enden an einem fixen Punkt innerhalb des Containerschiffs. Dieser bildet bezüglich Breite und Tiefe in etwa das Mittel der vorliegenden Containerslots. Damit wird von Seitwärtsbewegungen der Containerbrücken abstrahiert. An den Containerbrücken finden im realen System beim Löschen eines Schiffes Lascharbeiten an jedem Container statt. Diese werden nicht explizit modelliert. Die für das Laschen benötigte Zeit kann jedoch pauschal durch eine Verlangsamung der Hub- und Bewegungsgeschwindigkeiten an Laufkatze und Spreader der Containerbrücken im Modell umgesetzt werden.

Es wird für jede Modellkomponente eine feste Geschwindigkeit angenommen. Auf Beschleunigungs- und Bremsphasen wird verzichtet.

Weiterhin wird von einer Abbildung von Eisenbahngates abgesehen, da diese, genau wie die LKW-Gates, einen Portalkran als Ressource eine Zeit lang binden. Im Rahmen der zu treffenden Effizienzaussagen spielt die Unterscheidung der Gates also nur eine untergeordnete Rolle.

Im Präsentationsfilm von ZPMC [ZPMb] werden Tandem-Spreader verwendet, die zwei benachbarte 40 ft Container gleichzeitig transportieren können. Ebenso sind die Flat- und Lifting-Trolleys sowie die Terminal-Chassis dort auf den simultanen Transport zweier benachbarter Container gleichzeitig ausgelegt. Im vorliegenden Modell wird davon abstrahiert, da die Auswirkungen auf die Interaktionen der Betriebsmittel als gering eingeschätzt werden. Allerdings ist anzunehmen, dass der Einsatz von Tandem-Lifts einen erheblichen Einfluss auf zu treffende Effizienz- und Kapazitätsausagen hat. Dieser Einfluss wird im Rahmen der Diplomarbeit noch diskutiert.

4.1.3. Modellgrenzen, Dimensionen und Parameter

Aus den zu modellierenden Betriebsmitteln (Abschnitt 4.1.1) und den dabei gewählten Abstraktionen (Abschnitt 4.1.2) ergeben sich die Grenzen des Simulationsmodells. Containertransportaufträge sind die transienten Entitäten des Terminals. Sie beschreiben den Weg eines Containers durch das System Containerterminal und binden während ihrer Durchführung Betriebsmittel, wie etwa Krane, als persistente Entitäten des Modells. Dabei werden drei unterschiedliche Arten von Transportaufträgen ausgemacht. Seeseitige Aufträge beschreiben einen Containertransport zwischen einem Stack und einem Containerschiff. Sie untergliedern sich in „Import-Aufträge“ für das Entladen des Schiffs und „Export-Aufträge“ für das Beladen. Analog dazu beschreiben landseitige Im- und Exportaufträge das Be- und Entladen von LKW an den Gates. Eine dritte Gruppe bilden die „Restacking Aufträge“. Sie definieren das Umstapeln innerhalb des Containerterminals und dienen der Vorbereitung der übrigen Aufträge. Die Container werden in ihrer Abholungsreihenfolge vorgestaut und in die passenden Bereiche des Yards gebracht. Die Systemgrenzen sind dementsprechend genau die Punkte, die den Start bzw. das Ende eines Transportauftrages darstellen. Das sind zum einen die Containerterminal-Außengrenzen, landseitig die LKW an den Gates und seeseitig die Übergabepunkte der Containerbrücken auf den Schiffen. Zum anderen sind es innerhalb des Terminals die Übergabepunkte in den Containerblöcken der Stacking-Area.

Das zu erstellende Modell soll mit weitreichenden Parametrisierungsmöglichkeiten versehen werden, um mit möglichst vielen unterschiedlichen Versuchsaufbauten zu experimentieren und diese bezüglich ihrer Leistungsfähigkeit gegeneinander abzuwägen. Dabei

lassen sich verschiedene Typen von möglichen Parametrisierungen unterscheiden:

- *Parametrisierung des Modellaufbaus:* Es soll möglich sein, das Horizontalkrankonzept mit unterschiedlich vielen eingesetzten Betriebsmitteln an der Low-Frame-Bridge zu untersuchen. Dazu soll zum einen die Anzahl der eingesetzten Horizontalspuren wählbar sein. Es sind bis zu vier parallele Spuren möglich. Zum anderen soll für jede Spur die Anzahl der darauf arbeitenden Flat- und Lifting-Trolleys einstellbar sein, dabei sind bis zu vier Flat- und acht Lifting-Trolleys möglich.

- *Wahl technischer Parameter:* Es ist sinnvoll, den Aufbau des Beispielterminals mit unterschiedlichen Betriebsmitteln und verschiedenen technischen Spezifikationen zu untersuchen. Deswegen werden die für das Modell wichtigen technischen Merkmale wie Geschwindigkeit der Laufkatzen sowie Fahr- und Hubgeschwindigkeiten als veränderliche Parameter realisiert. So kann zum Beispiel überprüft werden, ob der Einsatz einer leistungsstärkeren Containerbrücke Effizienzgewinne für das Gesamtsystem mit sich brächte. Für die Komponenten des Horizontalkrankonzepts, Flat- und Lifting-Trolley, liegen keine technischen Spezifikationen vor, so dass diese Werte geschätzt werden müssen. Um hier eine Parametrisierung vorzunehmen, sollten unterschiedliche Annahmen getestet werden.

- *Experimentparameter:* Um den Containerterminalbetrieb entweder bezüglich eines in engen zeitlichen Grenzen gestellten Untersuchungsgegenstandes wie der Beladung eines einzelnen Schiffes oder über einen längeren Zeitraum wie der Untersuchung einer gesamten Wochenplanung inklusive aller Schiffsankünfte zu simulieren, muss die Experimentdauer sowie die Dauer des Experimenttraces und einer möglicherweise nötigen Debugperiode voreinstellbar sein.

- *Szenarioparametrisierung:* Die Einstellung der Szenarioparameter bedeutet die Festlegung der Schiffsankünfte und der Transportaufträge mit deren Eigenschaften bezüglich Typ und Anzahl. Des weiteren werden die beteiligten Betriebsmittel, Lagerblöcke und Containerschiffe definiert. Dabei sollen ähnliche Transportaufträge gebündelt in einem Schritt erzeugbar sein.

4.2. Dimensionierung des Modells

Beim vorliegenden Modell handelt es sich nicht um die Abbildung eines realen, bestehenden Systems, sondern um ein künstliches, erdachtes Containerterminal, das den Rahmen für die Untersuchung des Horizontalkrankonzeptes darstellt. Daher wird sich die Modellierung an bestehenden Containerterminals orientieren, um eine möglichst realitätsnahe Umgebung für die Komponenten des Horizontalkrankonzeptes zu schaffen. Wo es möglich ist, werden existierende Betriebsmittel mit ihren technischen Spezifikationen abgebildet. Gibt es diese nicht oder sind sie nicht bekannt, werden möglichst realistische Werte geschätzt. Als Ausgangsmodell dient dabei ein am ISL vorliegendes Containerterminalmodell.

Als Größe des Containerterminals wird ein Containerdurchsatz von 2,5 Mio TEU angenommen. Dieses entspricht etwa dem Umschlag des Containerterminals Burchardkai [HHLc] der HHLA [Mit].

Da im vorliegenden Modell von 20 ft Containern abstrahiert wird, (Abschnitt 4.1.2) ergibt sich ein Umschlag von 1,25 Mio Boxen pro Jahr. Bei einer angenommenen durchschnittlichen Verweildauer der Container im Stack von 4,5 Tagen werden im Mittel Containerblöcke mit einer Kapazität von 15.410 40 ft Containern benötigt.

Der Aufbau der einzelnen Containerblöcke wird an einen Film der Firma ZPMC angelehnt, in dem das Horizontalkrankonzept illustriert wird [ZPMb]. Dort werden jeweils Containerblöcke mittig durch die Spur eines Terminalchassis unterbrochen. Beide Blöcke werden inklusive der Spur von zwei Portalkranen komplett überspannt und bilden so als Doppelblock eine Einheit. Die Containerblöcke des Simulationsmodells wurden analog dazu abgebildet. Jeder Block besteht aus 6 nebeneinander liegenden Reihen von Containern die je 26 40ft Boxen lang und bis zu 4 Lagen hoch sind. Damit besitzt jeder Block eine Kapazität von

$$6 \text{ Reihen } * 26 \text{ Stapel} * 4 \text{ Lagen } = 624 \text{ Slots.}$$

Ein Doppelblock hat damit eine Kapazität von 1248 Boxen. Zielt man dann auf eine mittlere Auslastung des Stacks von 70% ab, so erhält man eine Kapazität von

$$1248 \text{ Slots} * 0.7 = 873,6 \text{ Slots.}$$

Ausgehend vom oben genannten mittleren Gesamtstellplatzbedarf von 15.410 ergibt sich

$$15.410 \text{ Gesamtstellplatzbedarf}/873 \text{ Slots } = 17,65 \text{ Doppelblöcke},$$

also 18 Doppelblöcke, um das Modell hinreichend groß zu dimensionieren.

Die Doppelblöcke werden in Analogie zum ZPMC-Film [ZPMb] mit zwei schienengeführten Portalkranen modelliert, die zum einen das Terminal-Chassis be- und entladen und zum anderen zwischen Stack und dem an der Stirnseite des Stacks gelegenen Gate Conatiner bei landseitigen Im- und Exporten umschlagen. Die Berthing-Area wird mit drei Liegeplätzen ausgestattet, an denen je nach Schiffsgröße bis zu acht Containerbrücken gleichzeitig arbeiten können. Die Low-Frame-Bridge wird derart dimensioniert, dass sie mit bis zu vier Spuren mit je bis zu vier Flat- und acht Lifting-Trolleys befahren werden kann.

Technische Parameter

Neben den weitreichenden Möglichkeiten zur Parametrisierung des Modells wird eine Standardparametrisierung angelegt, die Default-Werte für die einzelnen Betriebsmittel umfasst. Diese orientieren sich, soweit möglich, an realen Werten aus bestehenden Containerterminals. Insbesondere die technischen Daten der einzelnen Betriebsmittel sollen realistische Bedingungen zur Simulation der neuartigen Architektur liefern. Referenzen zu bestehenden Systemen finden sich dazu auf der Internetseite der Firma Siemens [Siea]. Die im Simulationsmodell abgebildete Containerbrücke orientiert sich an einer im Containerterminal Altenwerder [HHLb] eingesetzten Brücke der Firma ZPMC [Siec]. Diese besitzt eine Hubwerksgeschwindigkeit von 80 m/min beladen und 180 m/min unbeladen. Bei der Abbildung auf die Modellkomponente wird von dieser Unterscheidung abstrahiert und der Mittelwert von 130 m/min angenommen. Die Geschwindigkeit der Laufkatze wird anolog zum Referenzmodell mit 220 m/min modelliert. Die Portalkrane werden entsprechend des am Burchardkai [HHLc] eingesetzten Modells [Sieb] des Herstellers Kran Service Rheinberg, Demag abgebildet. Für die Hubgeschwindigkeit wird der Mittelwert zwischen beladenem (30 m/min) und unbeladenem (60 m/min) Hub gebildet. Dieser beträgt 45 m/min. Die Trolleygeschwindigkeit und die Fahrgeschwindigkeit werden unverändert mit 100 m/min bzw. 120 m/min übernommen.

Für die zentralen Komponenten des Horizontalkrankonzeptes liegen keine technischen Spezifikationen vor. Deshalb müssen einige Werte geschätzt werden. Dabei werden die Geschwindigkeiten der Terminal-Chassis und der Flat-Trolleys mit 18 km/h angenommen. Für den Lifting-Trolley werden Fahrgeschwindigkeiten von 60 m/min, Hubgeschwindigkeiten von 50 m/min und eine Dauer für das Drehen eines Containers um 90 Grad von 5 s zu Grunde gelegt. Für das Aufnehmen und Absetzen eines Containers werden ebenfalls je 5 s veranschlagt.

Bei der Verwendung des Simulationsmodells sollte dringend überprüft werden, ob sich diese Werte durch technische Spezifikationen oder bessere Schätzungen ersetzen lassen.

4.3. Eingesetzte Werkzeuge

Das Vorgehen bei der Modellerstellung, Durchführung von Simulationsexperimenten und Animation der Simulationsergebnisse lässt sich in fünf Schritte unterteilen, bei denen jeweils verschiedene Programme eingesetzt werden. (Abb. 4.1)

- *Erstellung von 3D-Komponenten in AC3D:* Es werden 3D-Komponenten modelliert, die in der Animation Komponenten des Realsystems repräsentieren.
- *Design eines Modell-Layouts mit Verdigo:* Dabei werden für alle Komponenten Position und Ausdehnung in einem dreidimensionalen Koordinatensystem festgelegt und die dazugehörenden 3D-Komponenten hinterlegt.
- *Erstellung des Simulationsmodells in Eclipse SDK und Netbeans IDE:* Bei der Erstellung des Simulationsmodells im Javaframework Desmo-J wird die Entwicklungsumbebung Eclipse SDK zur Modellierung der Simulationslogik eingesetzt. Das grafische Benutzerinterface wird in Netbeans IDE realisiert.
- *Durchführung eines Simulationsexperimentes:* Zur Durchführung eines Simulationsexperimentes müssen das Simulationsmodell als Java-Programm und das dazu gehörende Modell-Layout aus Verdigo vorliegen. Dabei wird zunächst ein Szenario erstellt oder geladen. Anschließend wird mit der vorgenommenen Parametrisierung ein Simulationslauf durchgeführt, der neben einigen statistischen Auswertungen eine Animationsdatei erzeugt.

- *Animation des Simulationsergebnisses:* Die während des Simulationslaufs erzeugte Animationsdatei kann in einem weiteren Schritt im 3D Realtime Animation Player Golem 3D abgespielt werden.

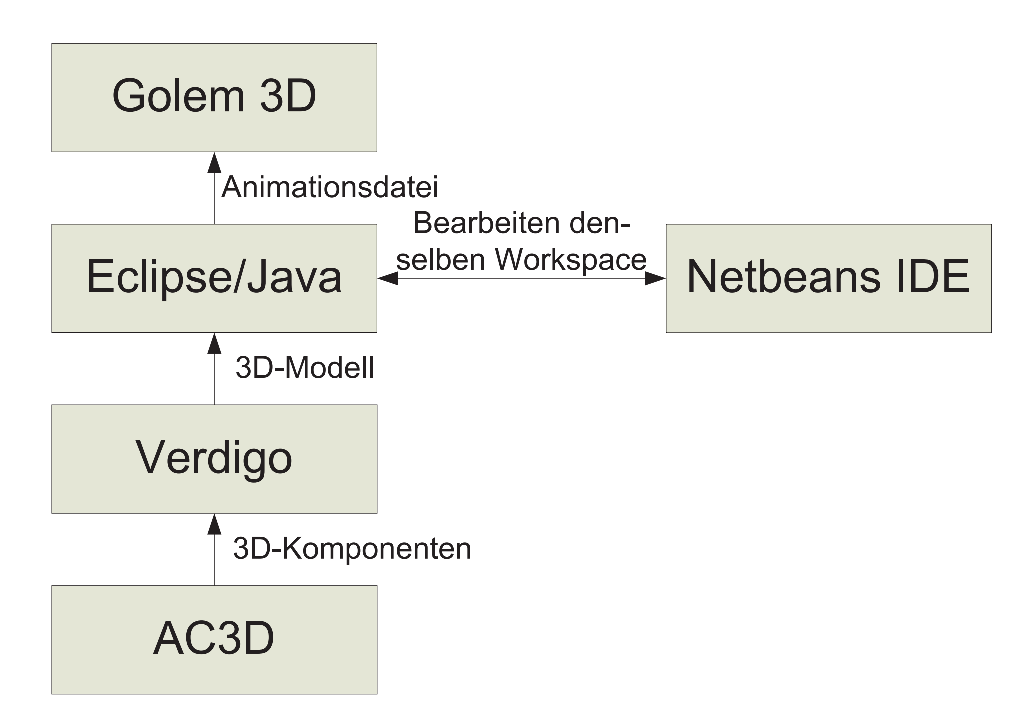


Abb. 4.1.: **Modellierung, Simulation und Animation in den unterschiedlichen Programmen**

4.4. Abbildung der Stau- und Stellplatzplanung

Gegenstand der vorliegenden Arbeit ist die Untersuchung des Horizontalkrankonzepts in einem Containerterminal. Dabei steht die Interaktion der Betriebsmittel und die daraus emergierenden Leistungskennziffern des Gesamtsystems im Fokus. Diese beschreiben zum einen absolute Gesamt- und Durchschnittswerte des Be- und Entladens von Containerschiffen und LKW sowie einzelner Containertransporte im System. Zum anderen bilden sie relative Werte gegenüber einem Referenzmodell. Die konkrete Stau- und Stellplatzplanung ist dabei ein Randaspekt. Ihre explizite Abbildung hätte nur einen relativ geringen Nutzen für die Aussagekraft der Ergebnisse des Simulationsmodells und hätte

darüber hinaus den Rahmen dieser Diplomarbeit gesprengt. Trotzdem hat sie einen erheblichen Einfluss auf das Verhalten des Modells, so dass sie über möglichst realistische Annahmen der modellierten Systembereiche implizit umgesetzt wird. Zum Beispiel könnte das Beladen eines Containerschiffs ausschließlich aus dem am dichtesten gelegenen Containerslot des am dichtesten gelegenen Stacks zu herausragenden Leistungskennziffern führen, entspräche aber keinen realistischen Annahmen für die Stellplatzplanung.

Die verschiedenen Implikationen der Stell- und Stauplanung werden beim Erstellen von Schiffsankünften und Transportaufträgen sowie der Parametrisierung der Auftragsvorbedingungen antizipiert. Sie sind teilweise direkt im Simulationsmodell realisiert, unterliegen jedoch auch der Verantwortung des Anwenders.

Dieser hat bei der Erstellung eines Szenarios darauf zu achten, dass er die Transportaufträge so anlegt, dass sie einer echten Stauplanung entsprechen könnten. Beispielsweise dürfen 1500 Container, die aus einem großen Containerschiff entladen werden sollen, nicht in einen einzigen Stack mit einer Kapazität von 1280 Containern eingestellt werden. Restacking-Aufträge sollten insbesondere die Containerblöcke betreffen, in denen ein hoher land- und seeseitiger Umschlag stattfindet. Des weiteren obliegt es dem Benutzer dafür zu sorgen, dass ein Bereich des Schiffes erst komplett entladen werden muss, bevor er wieder beladen werden kann. Die Auftrags-Eingabemaske stellt dafür den Parameter „Cycle Lenght“ zur Verfügung, der bestimmt, wie viele Im- oder Exportaufträge am Stück abgearbeitet werden müssen.

Die im Modell umgesetzten Implikationen der Stau- und der Stellplatzplanung betreffen die Zuordnung von Containerbrücken und Stacks für eine Gruppe von Aufträgen aus einer Liste. Wird ein Szenario vom Anwender so definiert, dass Aufträge gebündelt angelegt werden, ordnet das Modell die angegebenen Stacks und Containerbrücken zufällig gleichverteilt zu und antizipiert so eine gleichmäßige Ressourcennutzung durch die Stellplatzplanung.

Ausschließlich vom Modell gesteuert wird die Zuordnung bestimmter Containerslots zu einem Transportauftrag. Dabei wird bei seeseitigen Exportaufträgen der Platz gleichverteilt zufällig aus der vorderen Hälfte des Stacks zugeordnet und drückt so einen Vorstau durch die Stellplatzplanung aus. Bei den übrigen Auftragstypen werden die Containerslots über den gesamten Stack verteilt generiert und spiegeln so eine gleichmäßige

Nutzung desselben wieder.

Ein weiterer Punkt im vorliegenden Simulationsmodell, an dem beschrieben wird, wie die Stauplanung implizit im Modellverhalten berücksichtigt wird, ist die Parametrisierung, unter welchen Umständen Transportaufträge im Modell bearbeitet werden sollen. Dazu ist eine aus drei verschiedenen Regeln wählbar, von denen jede eine unterschiedliche Sichtweise darstellt, wie die Stau- und die Stellplatzplanung im Modell umgesetzt werden. Die Stellplatzplanung im Containerterminal bedingt die Reihenfolge, in der die Container in einem Containerblock gestapelt sind bzw. in welcher Reihenfolge sie gestapelt werden sollen. Die Stauplanung bedingt die Reihenfolge in einem Containerschiff. Je nachdem an welcher Stelle im Modell man die Stau- und Stellplatzplanung logisch realisiert sieht, stehen unterschiedliche Transportaufträge zur Durchführung bereit. Dabei ist jede dieser Regeln eine Abstraktion des realen Systemverhaltens und bildet unterschiedliche Teile desselben ab.

- *Strikte Ordnung:* Die strikte Einhaltung der Ankunftsreihenfolge der Transportaufträge bei ihrer Abarbeitung bildet die Einhaltung von Reihenfolge-Beziehungen explizit auf die Zugangssteuerung des Systems ab. Die Forderung einer strikten Ordnung ist gegenüber dem Realsystem allerdings zu stark. In Wirklichkeit bestehen zwischen einigen Aufträgen Abhängigkeiten, zum Beispiel zwischen einem Restacking-Auftrag und einer Entnahme des Containers für einen Weitertransport oder zwei Aufträgen, die denselben Containerslot im Schiff oder im Stack beschreiben. Andere Transportaufträge können jedoch in beliebiger Reihenfolge parallel abgearbeitet werden, je nachdem für welchen die nötigen Betriebsmittel zuerst verfügbar sind.
- *Greedy:* Diese Regel zum Systemeintritt von Transportaufträgen überprüft keine Reihenfolgebedingungen durch die System-Eingangskontrolle. Sie impliziert, dass wenn ein Auftrag abgearbeitet wird, eine optimale Stellplatzplanung seinen Transport zuvor ermöglicht hat. Eine explizite Abbildung der Reihenfolge findet nicht statt. Gegenüber dem Realsystem ist diese Annahme zu weitreichend.
- *Strikte Ordnung an den Containerbrücken:* Diese Regel bildet eine Mischform der beiden vorherigen. Sie erzwingt die Einhaltung der Auftragsreihenfolge für jede Containerbrücke, geht jedoch implizit davon aus, dass ein Auftrag, der durchgeführt wird, auf Seiten des Stacks durch passenden Vorstau ermöglicht wurde. Sie

> drückt damit aus, dass die Steuerung der Stellplatzplanung in der Lage ist im Terminal einen Vorstau zu erreichen, dieser aber nicht optimal für jeden Transportauftrag bezüglich der frei werdenden Betriebsmittel ist.

Welche der Systemzugangsregeln letztlich das Systemverhalten am besten abbildet, ist eine Entscheidung, die der Anwender zu treffen hat. Für die Untersuchung des Horizontalkonzepts im Rahmen dieser Arbeit wird die „Strikte Ordnung an den Containerbrücken“ verwendet, da sie die realistischste Annahme darstellt.

4.4.1. Modellierung von 3D-Modellen mit AC3D

Um einen Simulationslauf animieren zu können, müssen zunächst alle Modellkomponenten, die später Teile des grafischen Modells bilden, als 3D-Objekte vorliegen. Im Rahmen der vorliegenden Diplomarbeit wird dabei auf zahlreiche schon bestehende 3D-Objekte zurückgegriffen. Darüber hinaus werden jene Komponenten, die für das Horizontalkrankonzept spezifisch sind und damit noch nicht vorliegen, also der Flat- und der Lifting-Trolley und die Low-Frame-Bridge, mit der 3D Design Software AC3D erstellt.

Dabei werden in einem ersten Schritt alle geometrischen Körper, aus denen das spätere 3D-Objekt bestehen soll identifiziert. Für den Lifting-Trolley sind das drei Quader, einer als Deckplatte und zwei als Seitenwände, sowie neun Zylinder, sechs als Räder, einer als drehbare Plattform innerhalb der Deckplatte zur Aufhängung des Spreaders, und zwei als Motor bzw. Hubwerk.[ZPMc] In einem zweiten Schritt werden alle Teile in passender Größe zusammengefügt und überschüssige Bereiche entfernt. Dann werden die Oberflächen der einzelnen Teilobjekte entsprechend der Teile eingefärbt, die sie abbilden und mit einer Textur versehen, die dem Material aus dem sie bestehen, nämlich Stahl, nachempfunden ist (Abb. 4.2). Die Modellierung der übrigen neu zu erstellenden 3D-Modelle geschah analog.

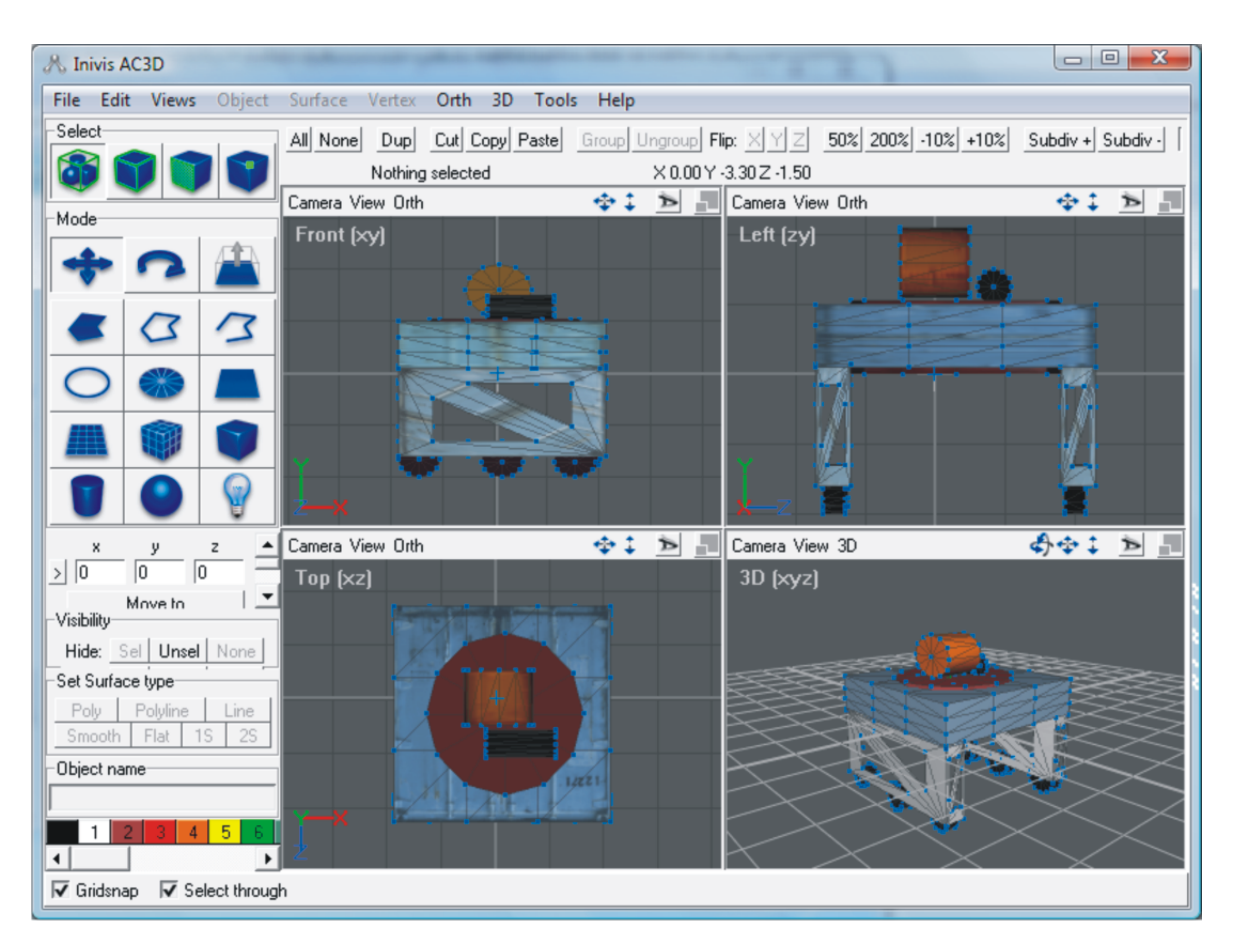


Abb. 4.2.: **Modellierung des 3D-Objektes eines Lifting-Trolleys in AC3D**

4.4.2. Erstellung des Modell-Layouts in Verdigo

Der 2D-Editor Verdigo [ISLc] dient der Anordnung von Modellkomponenten im dreidimensionalen Raum. Das ist insofern sinnvoll, als dass Desmo-J kein eigenes Raumkonzept besitzt, über das sich Größe, Lage, Ausrichtung und Anordnung von Modellkomponenten beschreiben lässt. Das in Verdigo erzeugte Modell-Layout lässt sich in Golem 3D [ISLa] betrachten. Es ist eine im Aufbau vollständige Animationsdatei, die alle Komponenten des Modells zum Zeitpunkt Null, also zu Beginn der Animation beschreibt. Über die Funktion „Export 3D“ ist es damit schon während der Modellierung des Layouts möglich aus Verdigo heraus ein Layout in Golem 3D zu betrachten.(Abb. 4.3)

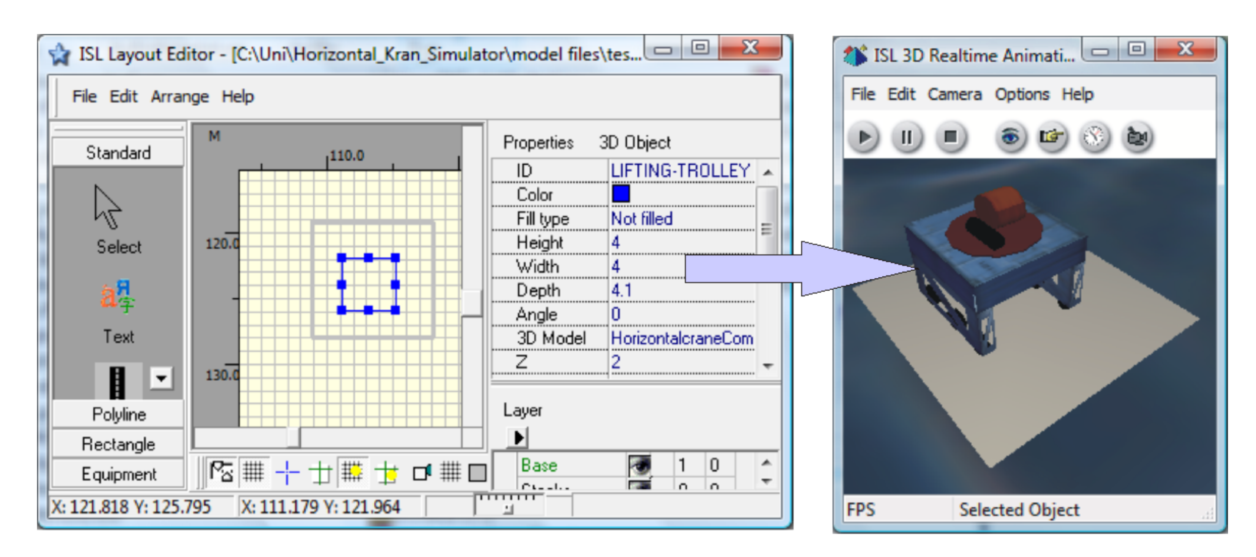


Abb. 4.3.: **Ansicht eines exportierten Verdigo-Layouts mit Golem 3D**

Um ein neues Modell zu erstellen, muss zunächst die Containerterminal-Gesamtfläche definiert werden (Abb. 4.6, 1). Sie grenzt die Landmasse des Containerterminals ab. Die nicht umschlossenen Bereiche bilden Wasser, das das Terminal umgibt.

Nun werden alle abzubildenden Betriebsmittel des Containertermials in das Modell-Layout eingefügt. Dazu wird zunächst jeweils eine Fläche des Typs „3D Objekt“ auf der Terminal-Area platziert. Ihr wird dann als Eigenschaft ein 3D-Modell zugeordnet. Dabei wird die zuvor gesetzte Grundfläche auf die Fläche des 3D-Modells angepasst. Die eingefügte Modellkomponente kann nun frei im Layout verschoben werden. Über die Eigenschaft „Autosize“ kann erzwungen werden, dass die Größe der Komponente der des 3D-Modells unveränderlich entspricht. Das ist in dem Fall sinnvoll, wenn das

3D-Modell ein reales Objekt in seiner Größe exakt abbildet. Modellkomponenten, die nicht über „Autosize“ auf eine bestimmte Größe festgelegt wurden, können, nachdem sie angelegt werden, auf eine bestimmte Größe skaliert werden. Dies geschieht, entweder indem direkt im Layout der Rahmen, der die Grundfläche der Komponente bildet, mit der Maus größer oder kleiner gezogen wird oder durch eine explizite Größenangabe in den Eigenschaften des 3D-Objektes. Sinnvoll ist dies dann, wenn z.B. unterschiedlich große Containerbrücken eingesetzt werden, die alle mit dem gleichen 3D-Modell abgebildet werden. Verdigo bietet des weiteren die Möglichkeit einzustellen, ob Objekte im 3D-Layout sichtbar sein sollen und ihnen ein weiteres Objekt als „Parent“ zu übergeben. Bei einer Bewegung oder Drehung des Parent-Objektes bewegen sich diese relativ dazu und können darüber hinaus noch selbstständig bewegt werden.

Im vorliegenden Modell wurde der Containertransport nicht durch eine echte Übergabe von 3D-Objekten realisiert (Abschnitt 4.1.2). Stattdessen wurden nur die Betriebsmittel, also Krane, Terminalchassis etc., persistent modelliert.

Dabei wird an jedes Betriebsmittel ein 3D-Objekt eines Containers über eine Parent-Beziehung gebunden. Dieser kann bei der Aufnahme eines Containers durch den jeweiligen Kran, Chassis etc. sichtbar gemacht werden. Wird der Container nun an ein weiteres Arbeitsgerät übergeben, so wird das dort gebundene Container-3D-Modell sichtbar gemacht. Der Container am Ausgangsgerät wird gleichzeitig auf „nicht sichtbar“ gesetzt, so dass der Eindruck einer echten Übergabe entsteht.

Eine Besonderheit stellen bei der Abbildung ins Modell-Layout die Containerblöcke der Stacking-Area dar. Im Gegensatz zu allen übrigen Komponenten des Containerterminals, die einzeln als 3D-Objekte abgebildet werden, gibt es hierfür einen eigenen Typ „Stack“. Er dient in der vorliegenden Modellierung ausschließlich zur Illustration von Lagerbereichen und als Platzhalter von Containerslots. Daher enthält das Stack-Modell keine 3D-Modelle von Containern, die einzeln adressier- und entnehmbar sind, sondern bildet ein großes Einzelobjekt. Damit wird zum einen während der Animation der Speicher der Grafikkarte entlastet, weil das Stack-Modell aus weit weniger Polygonen besteht, als die Summe der in ihm dargestellten Container-Modelle. Zum anderen stellt es für die Modellierung eine erhebliche Vereinfachung dar, keine explizite Belegung der Containerblöcke angeben zu müssen. Bei der Verwendung des Typs „Stack“ im Layout wird lediglich eine Grundfläche definiert und dem Block dann eine Kapazität zugewiesen. Diese besteht aus

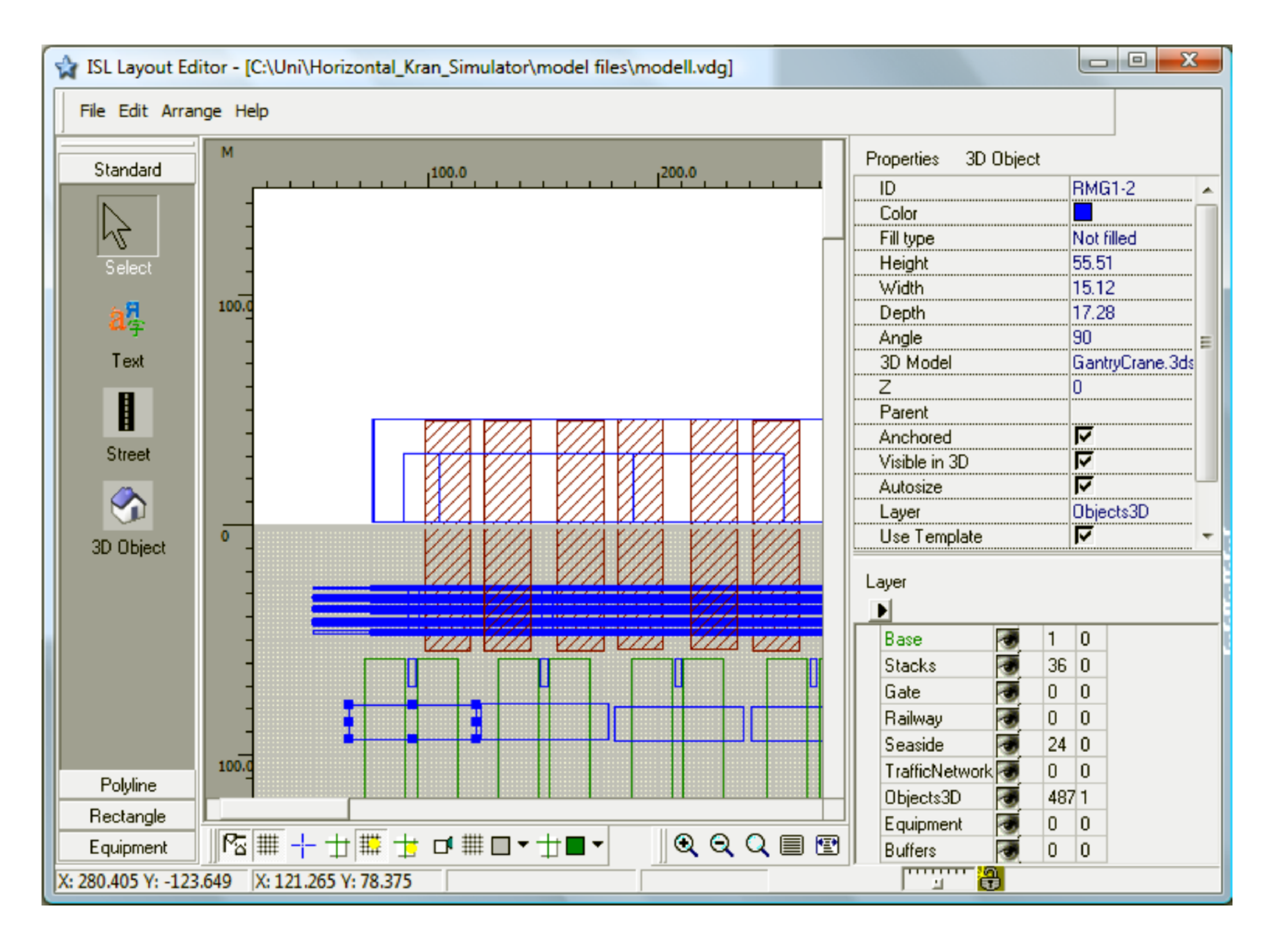


Abb. 4.4.: **Erstellung eines Modell-Layouts in Verdigo**

der Anzahl und Länge der Reihen sowie der Anzahl der Containerlagen. Über die Angabe des Füllgrades wird eine zufällige Belegung erzeugt. Da bei der vorliegenden Modellierung von expliziten Stau- und Stellplatzplänen abstrahiert wurde (Abschnitt 4.1.2) und damit nur gefordert wird, dass ein bestimmter Containerslot vorhanden, nicht jedoch unbedingt frei ist, entsprechen die im Typ Stack vorgenommenen Abstraktionen denen des Containerterminalmodells.

Templates zur Definition von Komponenten als Submodell

Verdigo bietet die Möglichkeit, Komponenten, die aus mehreren einzelnen 3D-Objekten bestehen, als Template zusammengefasst zu definieren und diese Templates in Form eines Submodells ein- oder mehrfach im Modell-Layout zu verwenden [ISLb].

Bei der Erstellung des Containerterminalmodells werden unter anderem alle Krane in Form von Templates modelliert, da dort jeweils verschiedene 3D-Modelle als Teil einer Komponente gemeinsam eine Einheit bilden. Am Beispiel des Portalkrans (Abb. 4.5) sollen hier die Definition und Verwendung eines Templates gezeigt werden. Templates werden in Verdigo auf der Basis eines 3D-Objektes angelegt. Im Hauptmodell kann später das Template, also hier das Portalkran-Template, an Stelle des einfachen 3D-Modells verwendet werden. Für den Portalkran bildet der Kranrahmen ohne jegliche bewegliche Teile das Basisobjekt. Ihm wird als Name ein Platzhalter, ein #-Symbol, zugewiesen, das bei einer späteren Verwendung im Hauptmodell mit einem eindeutigen Bezeichner überschrieben wird. Alle übrigen 3D-Objekte des Krans werden entweder direkt über eine Parent-Beziehung (Abschnitt 4.4.2) oder indirekt über eine Parent-Beziehung zu einem seinerseits mit dem Ausgangsobjekt assoziierten Objekt an das Hauptobjekt gebunden. Beim Portalkran wird in dieser Weise ein Trolley des Krans an den Rahmen gebunden, so dass er die Fahrbewegungen des Krans mitmacht und darüber hinaus auf der gesamten Breite des Rahmens über einer Containerreihe positioniert werden kann. Analog dazu ist der Spreader an den Trolley gebunden, so dass er sowohl die Kranbewegungen als auch die Positionierung des Trolleys nachvollzieht und sich zusätzlich auf und ab bewegen lässt. Genauso wird ein Container-3D-Modell am Spreader modelliert, das seinerseits allen Bewegungen des Spreaders folgt und zu unterschiedlichen Zeitpunkten auf sichtbar oder unsichtbar gesetzt werden kann.

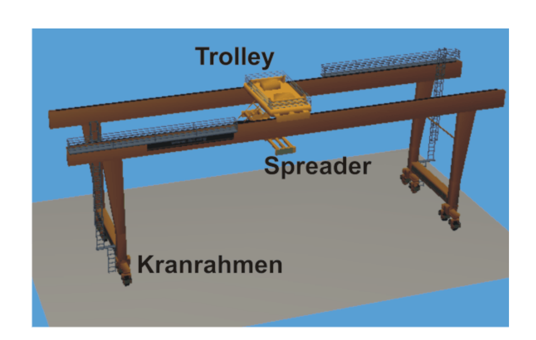


Abb. 4.5.: **Aufbau eines Portalkrans, Ansicht eines Modells in Golem 3D**

Bei der Anlage eines Templates müssen die Bezeichner der mit der Hauptkomponente verbundenen 3D-Modelle ein #-Symbol als Platzhalter für den späteren Template-Bezeichner im Modell-Layout sowie einen im Template eindeutigen Namenszusatz erhalten, so dass sich zusammen eine modellweite Eindeutigkeit ergibt. Beispielsweise wurde

für den Spreader der Bezeichner „-SP“ und für den Trolley „-TR“ gewählt, so dass im Gesamtmodell am ersten Portalkran des ersten Lagerblocks „RMG1-1“ der Name des Spreaders „RMG1-1-SP“ und der des Trolleys „RMG1-1-TR“ ergibt. Neben dem verringerten Modellierungsaufwand und leichterer Veränderbarkeit auch großer Modelle führt die Verwendung von Templates somit implizit auch zu einer einheitlichen Benennungspraxis.

4.5. Anordnung der Modellkomponenten in Verdigo

Im Verdigo-Modell-Layout entspricht die Anordnung der unterschiedlichen Modellkomponenten in Teilen dem konkreten Szenario eines Experiments. Die Gate-Area, die Stacking-Area, inklusive der Blöcke und der Portalkrane, sowie die Terminalchassis wurden dabei in ihrer Anzahl und Position so modelliert, dass sie für jedes mögliche Experiment die exakt gleiche Ausgangssituation liefern.

Die übrigen Komponenten, also die Spuren der Low-Frame-Bridge, die Flat- und die Lifting-Trolleys und die Containerbrücken werden im Layout in der für ein Szenario maximalen Anzahl angelegt. Ihre Anzahl innerhalb eines Szenarios ist später Teil der Parametrisierung, so dass unter verschiedenen Prämissen simuliert werden kann. Ähnlich werden die unterschiedlichen Schiffstypen realisiert. Dazu werden im Layout an den drei Berths alle innerhalb des Simulationsmodells möglichen Schiffe angelegt (Abb. 4.6, 7). Wenn während eines Simulationsexperimentes über einen Zeitraum ein Schiff an einem Liegeplatz liegt, wird ein entsprechendes Modell im Rahmen der Animation in der entsprechenden Berth über seine Eigenschaften sichtbar gemacht.

Der Gesamtaufbau des CT-Modells lässt sich in Abbildung 4.6 nachvollziehen. Dort befinden sich die unterschiedlichen Containerschiffe (7) jenseits des Terminalbereichs (1). Um die verschiedenen Schiffstypen abzubilden, lassen sich zwei Feeder-Schiffe mit 100 m, und 150 m Länge und drei Containerschiffe mit einer Länge von 200, 280 und 300 m animieren.

Diese können je nach Schiffstyp von bis zu acht Containerbrücken (4) be- und entladen werden. Die Brücken überspannen die Containerschiffe zur Gänze und werden mit einer Höhe von 50 m und einer Gesamtlänge von 100 m dimensioniert. Sie sind analog

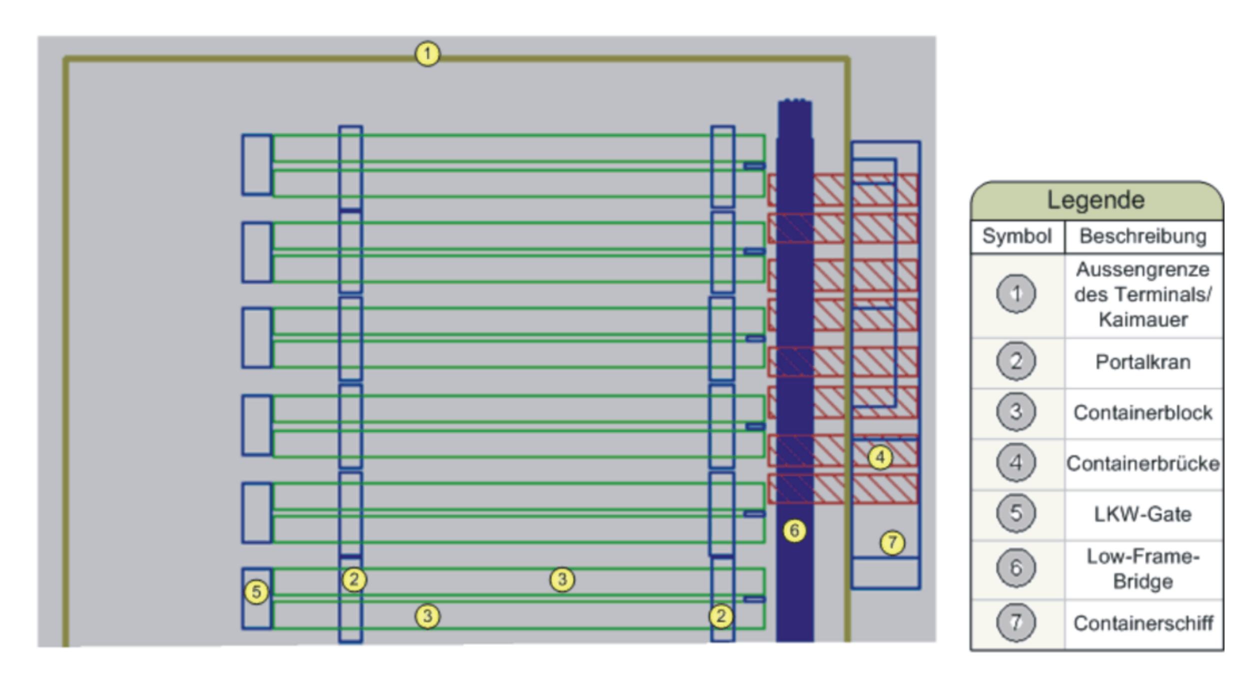


Abb. 4.6.: **Anordnung der unterschiedlichen Containerterminal-Komponenten; Abbildung eines Drittels des Gesamtmodells**

zu den Containerschiffen, gemäß der Parametrisierung, in der Animation sichtbar oder unsichtbar.

Die Low-Frame-Bridge (6) wird ebenfalls komplett unterhalb der Containerbrücken angelegt, so dass jede Spur mit den auf ihr beweglichen Flat-Trolleys vom Spreader der Brücke zu einer Containerübergabe erreichbar ist. Die Höhe der Low-Frame-Bridge wird mit sieben Metern so gewählt, dass sie von beladenen Terminal-Chassis (8) unterfahren werden kann und diese von den auf der Low-Frame-Bridge fahrenden Lifting-Trolleys be- und entladen werden können.

Der Aufbau des Stacks lässt sich mit dem des Containerterminals Altenwerder [HHLb] der HHLA vergleichen (Abb. 2.1). Genau wie in der vorliegenden Modellierung sind dort die langen Containerblöcke (3) charakteristisch, die jeweils mit zwei Portalkranen (2) bestückt sind. Sie interagieren, in Analogie zum CT-Altenwerder, terminalintern mit Chassis ohne eigene Hubfähigkeit sowie den LKW-Gates (5) an den landseitigen Stirnseiten der Stacks. Dort wird jeweils die Be- und Entladung der LKW durch einen Portalkran des Stacks vorgenommen.

Die Modellierung relativ langer Containerblöcke, 26 Container à 40 ft, insgesamt etwa 330 m, schafft zum einen eine Analogie zum Modell der Architektur im ZPMC-Präsentationsfilm [ZPMb], zum anderen wird so versucht, die mittlere Ressourcenbindung eines Transportauftrages über die Low-Frame-Bridge möglichst ausgeglichen auf die Betriebsmittel der Low-Frame-Bridge und denen der Stacking-Area zu verteilen und letztlich durch die kompakte Stapelung eine gute Raumausnutzung zu erreichen. Die Dimensionierung der Stacks mit jeweils zwei Portalkranen wird als sinnvoll erachtet, da nicht nur weite Wege innerhalb des Stacks zurückzulegen sind, sondern auch weil starker landseitiger Verkehr am Gate zur langen Bindung einer Kran-Ressource führen kann.

4.6. Beschreibung der Modell-Layoutdatei

Das Abspeichern eines Modell-Layouts in Verdigo erzeugt eine Reihe von Dateien, die neben der Möglichkeit das Layout wieder in Verdigo zu laden, eine Ansicht in Golem 3D erlauben. Außerdem wird eine XML-Datei, die alle Modell-Komponenten mit ihren Bezeichnern und definierten Attributen enthält, angelegt.

Die erweiterbare Auszeichnungssprache XML[1] dient der strukturierten Gliederung von Texten und Daten ([Ull09], Kap. 15). Sie definiert den formalen Aufbau von Dokumenten als Baumstruktur und ermöglicht Dank ihrer Erweiterbarkeit die eigenständige Formatierung von Daten und ihren Attributen. Die Typisierung von Daten geschieht dabei über die Elemente des Baumes durch ein Einschließen in eckige Klammern[2]. Über den Typ der Daten, also den Namen des sie umschließenden Elements, ergibt sich beim Auslesen, wie mit ihnen verfahren wird. Die durch ein Element beschriebenen Daten können entweder aus einer Zeichenkette bestehen oder ihrerseits rekursiv Datenelemente enthalten. Weiterhin können Elemente über Attribute genauer bestimmt werden. Diese besitzen einen Bezeichner und einen Wert.

XML-Dateien, die für eine Darstellung in Golem 3D vorgesehen sind, tragen die Endung „.ani“. Eine vollständige Definition der verwendbaren Elemente findet sich in der Golem

[1] XML: engl., Extensible Markup Language
[2] engl. tags

3D Schnittstellenspezifikation [ISL08]. Die für die Verarbeitung durch das Simulationsmodell benötigte Layout-Datei trägt den Namen „[Name des Modells]-objects.ani“. Sie beinhaltet alle im Layout definierten Komponenten mit ihren Eigenschaften in Form einer bestimmten Struktur (Abb. 4.7).

- *Version* (Zeile 1): Im Element „Version“ wird die verwendete XML-Version angegeben.
- *golem3d_animation* (Zeile 2): Das Element „golem3d_animation“ ist der Wurzelknoten der Golem 3D Animationsdatei.
- *caption* (Zeile 3): Die Caption ist der Name des Typs der vorliegenden Datei.
- *include* (Zeile 4-6, sowie 21): Weitere Dateien werden unter dem „include-Element“ die in die Struktur eingefügt. Das ist unter anderem die „[Bezeichner]-Modell.ani“ (Zeile 4), in der die automatisch generierten Schlüssel eines 3D-Modells (Zeile 15), mit dem Pfad der 3D-Modell-Datei verknüpft werden. Des weiteren führt das Attribut „replace“ dazu, dass die als Template erstellten Komponenten eines Submodells (Abschnitt 4.4.2, Zeile 5-6) an einen Bezeichner gebunden in das Modell eingefügt werden. Das Einfügen der Datei „Extra.ani“ (Zeile 21) fügt dem Modell weitere Elemente, in diesem Fall die Texturen für Wasser und Himmel sowie die Informationen, ob und wann welche Containerschiffe, Containerbrücken sowie die Spuren der Low-Frame-Bridge und ihre Betriebsmittel sichtbar sein sollen.
- *viewport* (Zeile 8): Ein „Viewport“ umfasst alle in der Animation wiedergegebenen Zeitpunkte.
- *time* (Zeile 9): Über das Attribute „value“ wird ein Zeitpunkt spezifiziert, an dem ein Ereignis in der Animation stattfindet.
- *object* (Zeile 10): Durch die Attributierung mit einer eindeutigen „id“ wird angegeben, welches Objekt durch das Ereignis verändert wird. Eine besondere Form ist dabei die Zuordnung eines *model*s, (Zeile 15) einer Position im dreidimensionalen Koordinatensystem *origin* (Zeile 11, 16) und der Ausrichtung relativ zum 3D-Modell *angle* (Zeile 12, 17). Dadurch wird das Objekt ab dem Zeitpunkt, an dem das Ereignis beschrieben wird, angezeigt. Wird eine Komponente zuvor als Template eingefügt, wird kein Modell mehr angegeben.

```
<?xml version="1.0" encoding="utf-8" ?>
<golem3d_animation>
    <caption>3D Animation Objects file</caption>
    <include> %xmlpath%beispiel-Model.ani</include>
    <include replace="#=LIFTING-TROLLEY">%AppPath%mymedia\include\templates\
        HorizontalcraneComplete\HorizontalcraneComplete-Objects.ani</include>

    <viewport id="vp1">
        <time value="0.0">
            <object id="LIFTING-TROLLEY" >
                <origin>78 -36 0</origin>
                <angle>0 0 0</angle>
            </object>
            <object id="TRUCK"  autorotate="1" >
                <model>{1DFFBAAC-B5B3-4743-B7DD-8310DF4AC073}</model>
                <origin>108.19 -29.851 0</origin>
                <angle>0 0 0</angle>
            </object>
        </time>
    </viewport>
<include>%xmlpath%extra.ani</include>
</golem3d_animation>
```

Abb. 4.7.: **Beispiel einer Modell-Objects.ani Datei, angezeigt in Notepad++**

5. Aufbau der Modellsteuerung in Java mit Desmo-J

Die Steuerung des Modells lässt sich in zwei Bereiche einteilen. Der eine Bereich bildet eine globale Steuerung, die alle in einem Zeitpunkt vorliegenden Transportaufträge zunächst in einer Warteschlange zurückhält und sie erst, sobald die nötigen Betriebsmittel verfügbar sind, an die beteiligten Modellkomponenten übergibt und diese aktiviert. Die Steuerung der am Transport beteiligten Modellkomponenten und deren Interaktionen bilden den anderen Bereich der Steuerung des Systems. Dort interagieren die Komponenten des terminalinternen Verkehrs und führen den eigentlichen Containertransport durch. Dieser Bereich wurde mit Simulationsprozessen umgesetzt.

Die Verwendung von Prozessen und Ereignissen in einem hybriden Modellierungsstil verbindet verschiedene Vorteile des prozess- und des ereignisorientierten Modellierungsstils. Die Abbildung der Betriebsmittel des Terminals als Simulationsprozess erscheint als sinnvoll, weil dort die Steuerungslogik einer Komponente gekapselt und an die Entität gebunden vorliegt. Die Modellierung der der Ankünfte der Transportaufträge am System ihrer Systemeintritte als Ereignis betont, das von Seiten des Transportauftrages kein eigenes aktives Verhalten vorliegt. Außerdem wird für jeden `SimProcess` ein eigener Java-Thread verwendet, so dass größere Anzahlen gleichzeitig im System befindlicher Transportaufträge auf Grund beschränkter Speicher-Ressourcen nicht abbildbar wären.

Innerhalb des Simulationsmodells wurden Wahrscheinlichkeitsverteilungen zur Abbildung nicht deterministischen Systemverhaltens verwendet. Dieses umfasst die Zuordnung von Containerbrücken, Stacks und Containerstellplätzen sowie der Modellierung von LKW- und Schiffsverspätungen.

5.1. Einlesen des Modell-Layouts

Um Bewegungen zwischen Punkten des Containerterminals in der Simulation abbilden zu können, benötigt man ein Raummodel des Terminals. Dieses liegt in der Layout-Datei vor und muss in das Simulationsmodell überführt werden, um die Positionen und Abstände der abgebildeten Komponenten bekannt zu machen. Dazu wurde das XML-Dokument, in dem das Layout definiert wurde, in das Modell eingelesen.

Für den Zugriff auf die XML-Datei des Layouts wurde die „jDom"-Api verwendet. Mit ihr wird zu Beginn einmalig eine DOM-Repräsentation der Datei erzeugt. DOM steht für „Document Object Model" und ist eine Schnittstellenspezifikation für XML. Durch sie wird das XML-Dokument komplett in seiner Baumstruktur betrachtet. Ausgehend vom Wurzelknoten des Dokumentes werden Methoden zum Lesen und Manipulieren von Daten, Attributen, deren Werten, sowie dem Zugriff auf abgeleitete Knoten etc. beschrieben ([Har02], Kap. 9-12). jDom setzt die DOM-Spezifikation als Java Api um und bildet die DOM-Schnittstelle auf Java-Methoden ab. Um einen Zugriff auf alle Teile des Dokuments zu erlauben, wird das gesamte Dokument zunächst in den Speicher geladen. Die interne Java-Repräsentation bildet dabei alle Daten eines Elementes über eine Liste[1] ab, so dass Kind-Elemente rekursiv Listen von Listen bilden ([ME07], S. 243-287).

Zur Einbindung in das Simulationsmodell wird bei dessen Initialisierung ein Ojekt der Klasse `XMLManipulatorSingleton` erzeugt, das zum Auslesen der Layoutdatei, sowie der Erstellung der Animationsdatei dient. Die Realisierung als Singleton-Pattern erzwingt dabei, dass modellweit nur ein Objekt der Klasse vorliegt und so sichergestellt wird, dass die Daten immer aus derselben Layout-Datei eingelesen und in dieselbe Animationsdatei geschrieben werden. Dieses Objekt stellt die Methode:

```
public DimRepresentation getOrigin(String id)
```

zur Verfügung, mit der bei Angabe eines in Verdigo vergebenen Bezeichners dessen Koordinaten im dreidimensionalen Raum in Form eines Objektes vom Typ `DimRepresentation` zurückgibt, das die Positionsinformation enthält.

[1] java.util.List

5.2. Steuerung modellinterner Abläufe durch Simulationsprozesse

Alle persistenten Betriebsmittel des Containerterminals, die am Containertransport beteiligt sind, wurden als `SimProcess` modelliert. Dazu wurde für alle unterschiedlichen Geräte jeweils ein eigener von `SimProcess` abgeleiteter Typ definiert. Dieser enthält in seiner `lifeCycle()`-Methode alle Aktivitäten, die seinen Anteil am Transport ausmachen inklusive der Interaktionen mit anderen Modellkomponenten.

Die Simulationsprozesse bilden das Verhalten der jeweiligen Komponente in ihrer Steuerung ab. Bei ihrer Erzeugung im Simulationsmodell wird den Prozessen im Konstruktor jeweils explizit die ID ihrer Komponente aus dem Verdigo-Layout übergeben. Dabei wird wiederum ein ihnen entsprechendes `ModelComponent`-Objekt angelegt, das die Position und Bewegung der Komponente im Raum des Modells kapselt. Ihnen entsprechend bedeutet, dass ein `ChassisProcess` eine `ChassisComponent` und ein `QuayCraneProcess` eine `QuayCraneComponent` anlegt usw.

5.2.1. Modellkomponenten zur Kapselung konkreter Positionen und Bewegungen

Diese Modellkomponenten bilden die Schnittstelle zwischen den Simulationsprozessen und der Animationsausgabe. Sie kapseln außerdem die konkrete Position und die Art und Weise, wie Bewegungen der Komponente und ihrer Subkomponenten zwischen zwei Punkten auf dem Containerterminal stattfinden.

Dazu wurde zunächst die abstrakte Oberklasse `ModelComponent` angelegt. Dort ist definiert, dass jeder Komponente bei ihrer Erzeugung die ID der ihr entsprechenden Komponente des Modell-Layouts übergeben wird. Mit ihr wird über die `getOrigin(String id )`-Methode im `XMLManipulatorSingleton`-Objekt die Anfangsposition, sowie die ID des Containers bestimmt, der in der Animation über eine Parent-Beziehung (Abschnitt 4.4.2) mit der Komponente verbundenen ist[2]. Weiterhin wird eine abstrakte Methode zur Bewegung der Komponente deklariert, die in den Unterklassen mit den entsprechenden

[2] [Bezeichner]+"-CT40"

Geschwindigkeiten implementiert wird, sowie Methoden zum sichtbar und unsichtbar Machen von Containern und dem Bestimmen der Entfernung zu anderen Komponenten.

In den konkreten Implementationen der Klasse `ModelComponent`, zum Beispiel `QuayCraneComponent` oder `ChassisComponent`, werden dann etwaige Erweiterungen um die Subkomponenten Trolley ([Bezeichner]+“-TR“) und Spreader ([Bezeichner]+“-SP“) vorgenommen, die die verbundenen Subkomponenten in der Animation beschreiben.

Hier wird auch die Implementation der `moveTo`-Methode sowie die möglicher weiterer Methoden für das Bewegen von Spreadern und Trolleys vorgenommen (Abschnitt 5.1). Dabei wird der Methode immer ein Ziel und ein Zeitpunkt übergeben, aus dem, ausgehend von der momentanen Position der Komponente, die Strecke und dann aus Strecke und Geschwindigkeit die Fahrdauer errechnet wird. Anschließend wird die Information über das `XMLManipulatorSingleton`-Objekt in die Animationsdatei geschrieben, die aktuelle Position in der Komponente aktualisiert und dem Simulationsprozess die Dauer der Bewegung zurückgegeben.

```
1  public double moveTo(double timeValue, DimRepresentation target) {
2
3      //Bestimmung des Übegabepunktes auf der Spur des Chassis
4      DimRepresentation targetPoint = new DimRepresentation (this.getCurrentPosition()
5          .getXvalue(),target.getYvalue(), this.getCurrentPosition().getZvalue());
6
7      double distance = this.getCurrentPosition().getDistance(targetPoint);
8      double duration = distance * (1/parameters.getChassisSpeed());
9
10     xmlManipulatorSingleton.moveObject(timeValue, duration, id,
11         this.getCurrentPosition().getPointAsString(), targetPoint.getPointAsString());
12
13     this.currentPosition = targetPoint;
14     return duration;
15 }
```

Listing 5.1: **Konkrete moveTo-Methode der Klasse ChassisComponent**

Neben einfachen Bewegungen stellen die `ModelComponent`-Objekte auch aggregierte Bewegungen zur Verfügung. Diese setzen sich aus mehreren Teilschritten in der Visualisierung zusammen, bilden in der Simulation jedoch logisch einen einzigen Schritt. Ein

Beispiel dafür ist in der `LiftingTrolleyComponent` die Bewegung zum Absetzen eines von einem Flat-Trolley übernommenen Containers auf ein unter ihm wartendes Terminal-Chassis. Während sie auf der Ebene des `LiftingTrolleyProcess` einen nicht unterbrechbaren logischen Schritt darstellt, setzt sie sich in der Komponente und damit auch in der Animation aus einem ersten Absenken, einer Drehung und der weiteren Absenkung auf das Chassis zusammen.

5.2.2. Modellierung von Interaktionen von Simulationsprozessen

Das Verhalten einer Modellkomponente über die Zeit wird durch einen Simulationsprozess abgebildet. Dieser interagiert mit den Prozessen anderer Komponenten und realisiert zeitkonsumierende Tätigkeiten. Abschnitt 3.2.

Im vorliegenden Simulationsmodell bestehen alle Zeit verbrauchenden Tätigkeiten immer aus zwei Schritten. Zunächst wird die Tätigkeitsmethode an dem Prozess zugehörigen `ModelComponent`-Objekt aufgerufen, z.B. ruft ein `ChassisProcess` an seiner `ChassisComponent` die Methode

```
moveTo(double timeValue, DimRepresentation target)
```

auf. Dort wird die Position der Komponente aktualisiert, das Bewegungsereignis an die Animation weitergegeben und die Dauer der Bewegung an den Simulationsprozess zurückgegeben, der daraufhin per `hold()` die Bewegung im Simulationsmodell realisiert.

Eine explizite Aktivierung der die Betriebsmittel abbildenden Simulationsprozesse wird nur beim Systemeintritt eines Transportauftrages in das System verwendet. Dort wird im Vorfeld sichergestellt, dass der angesprochene Prozess passiv wartend ist.

Prozesssynchronisation mit WaitQueues

An allen anderen Stellen, zum Beispiel der Übergabe eines Containers von einer Containerbrücke an einen Flat-Trolley, ist aus Sicht des `QuayCraneProcess`es nicht allgemein erkennbar, welches Gerät den Übergabepunkt zuerst erreicht und ob zum Zeitpunkt des

Erreichens des Übergabepunktes durch den `QuayCraneProcess` der `FlatTrolleyProcess` dort schon passiv wartet, also aktiviert werden kann. Deswegen wird überall dort, wo eine Synchronisation zweier Betriebsmittel benötigt wird, das von Desmo-J bereitgestellte Konstrukt `WaitQueue` verwendet. Eine `WaitQueue` dient der impliziten Synchronisation von Prozessen. Sie besteht intern aus zwei Warteschlangen. Sobald sich in beiden ein Prozess befindet, wird eine Kooperation durchgeführt, die den gemeinsamen Aktivitäten entspricht, in diesem Fall der Containerübergabe. Anschließend erfolgt implizit eine Aktivierung der beiden beteiligten Prozesse, so dass sie die Programmkontrolle zurückerhalten.

Ressourcenzugriff über Warteschlangen

Das Warten mehrerer gleicher oder ähnlicher Prozesse auf ein Ereignis wird über Warteschlangen, in Desmo-J `ProcessQueue`, realisiert. Sie berücksichtigen die Reihenfolge der Prozesse, zum Beispiel bezüglich der Dauer, wie lange der einzelne Prozess schon auf das Ereignis wartet. Die Warteschlange wird durch eine Liste beschrieben, bei der standardmäßig das „First In First Out"-Prinzip Anwendung findet. Darüber hinaus können alternative Zugriffsformen über die Definition von Bedingungen realisiert werden.

Während bei den übrigen Betriebsmitteln des Containerterminals jeweils nur eine exklusive Nutzung ab dem aktuellen Simulationszeitpunkt garantiert werden kann, können von einem Lifting-Trolley mehrere Transportaufträge eine zukünftige Nutzung anfordern. Der Lifting-Trolley arbeitet diese sequenziell ab. Um der Ressource gegenüber das Warten kenntlich zu machen und die Anfragereihenfolge auf den Prozess bei der Abarbeitung der Aufträge einzuhalten, wurde hier das Konstrukt `ProcessQueue` verwendet. Im vorliegenden Modell ist es deswegen möglich einen Containertransport zu beginnen, auch wenn der dazu gehörende Lifting-Trolley als Ressource noch nicht verfügbar ist.

Neben den oben genannten Funktionen einer Warteschlage zur Organisation seiner Mitglieder erlaubt die Implementation in Desmo-J weitreichende statistische Auswertungen. Diese umfassen die zeitbezogene Kennzahl zur durchschnittlichen Wartezeit, sowie kapazitätsbezogene Kennzahlen der aktuellen, durchschnittlichen und maximalen Warteschlangenlänge sowie die Anzahl der Elemente, die die Warteschlange schon verlassen haben.

5.3. Prozessinteraktionen zur Steuerung eines Transportauftrages

Programmintern beginnt jeder Containertransport durch das Containerterminal mit einem Systemeintrittsereignis. Dabei wird der Transportauftrag allen beteiligten Prozessen als Objekt übergeben. Diese werden dabei mit Ausnahme des Lifting-Trolleys reserviert und aktiviert. Beim Lifting-Trolley wird zunächst überprüft, ob dieser zuvor noch einen anderen Transportauftrag durchführt. Ist dies nicht der Fall, ist er also passiv wartend, so wird auch er aktiviert. Zusätzlich wird für den Flat-Trolley der Bereich seiner Spur auf der Low-Frame-Bridge reserviert, den er für den Transportauftrag zu befahren hat. Aus dem Auftrags-Objekt entnehmen die Prozesse, mit welchen anderen Modellkomponenten sie zur Durchführung des Transports kooperieren müssen und welche Rolle sie innerhalb des Transportes spielen. Dazu wurden innerhalb der `lifeCyle()`-Methode des Prozesses Fallunterscheidungen vorgenommen, die die unterschiedlichen Verhaltensweisen für alle Auftragstypen und Rollen während eines Transportauftrages spezifizieren. Die Unterschiede zwischen den Verhaltensweisen bezüglich verschiedener Auftragstypen lassen sich so verstehen, dass zum Beispiel eine Containerbrücke bei einem Export-Auftrag zunächst auf die Synchronisation mit dem Flat-Trolley warten muss und dann den Container auf das Schiff transportiert, während es bei einem Import-Auftrag genau anders herum ist. Durch die Rolle innerhalb desselben Auftragstyps wird unterschieden, ob ein Portalkran bei einem Restack-Auftrag derjenige Kran ist, der den Container auslagert oder jener, der ihn einlagert.

Durchführung eines Transportauftrages

Wie ein konkreter Containertransport durch das Terminal durchgeführt wird und in welcher Reihenfolge die unterschiedlichen Simulationsprozesse beteiligt sind, ergibt sich also aus der Spezifikation der beteiligten Simulationsprozesse und der Spezifikation der Steuerung in den Kooperationsobjekten.

Seeseitige Transportaufträge

Ein seeseitiger Import-Auftrag beschreibt den Transport eines Containers von einem Containerschiff an eine vorgegebene Stelle in der Stacking-Area. Beteiligt sind dabei eine Containerbrücke, ein Flat- und ein Lifting-Trolley, ein Chassis und ein Portalkran.

Mit ihrer Aktivierung zur Durchführung des Import-Auftrags führen alle beteiligten Prozesse mit Ausnahme des `QuayCraneProcess`es eine Bewegung hin zum Übergabepunkt durch, an dem sie den Container von ihrer Vorgängerkomponente des Containertransports erhalten werden und fügen sich dort in eine Master-Slave-Warteschlange ein und warten.

Der Prozess der Containerbrücke hebt mittels Trolley- und Spreader-Bewegungen einen Container vom Schiff und fährt diesen an den Synchronisationspunkt oberhalb der Low-Frame-Bridge. Dort fügt er sich in die `quayCraneFlatTrolleyCooperation`-Warteschlange ein. Befinden sich dort sowohl der `QuayCraneProcess` als auch der `FlatTrolleyProcess`, wird ein an die Warteschlange gebundenes `quayCraneFlatTrolleyHandover`-Objekt ausgeführt, das die Kooperation beider Prozesse spezifiziert, also das Absenken des Containers durch den Spreader der Containerbrücke auf den Flat-Trolley. Dann wird der Container vom Spreader gelöst und vom Flat-Trolley aufgenommen. Intern wird dies durch ein unsichtbar Machen des 3D-Container-Objekts am Spreader der Containerbrücke und ein sichtbar Machen auf dem Flat-Trolley sowie einer Zeitkonsumption für die Übergabe realisiert. Anschließend wird der Spreader der Brücke angehoben und die Kooperation beendet, wodurch beide Prozesse die Kontrolle zurückerhalten. Der `QuayCraneProcess` verändert seinen Status auf „unbeschäftigt“ und informiert die zentrale Terminal-Zugangssteuerung, den `OrderAllocator`, dass ihm neue Transportaufträge zugewiesen werden können (Abb. 5.1).

Der Flat-Trolley legt auf seiner Spur der Low-Frame-Bridge den Weg zur Übergabestelle mit dem Terminalchassis zurück und wartet dort in einer weiteren `WaitQueue` auf eine Kooperation zur Entladung durch einen Lifting-Trolley. Sobald beide den Übergabepunkt erreicht haben, sich also in die `WaitQueue` eingefügt haben, wird der Container im Rahmen einer Kooperation mit dem Lifting-Trolley vom Flat-Trolley gehoben. Der Flat-Trolley verlässt den Arbeitsbereich des Lifting-Trolleys, beendet seine Reservierung des Spurbereichs und stellt sich für weitere Transportaufträge zur Verfügung.

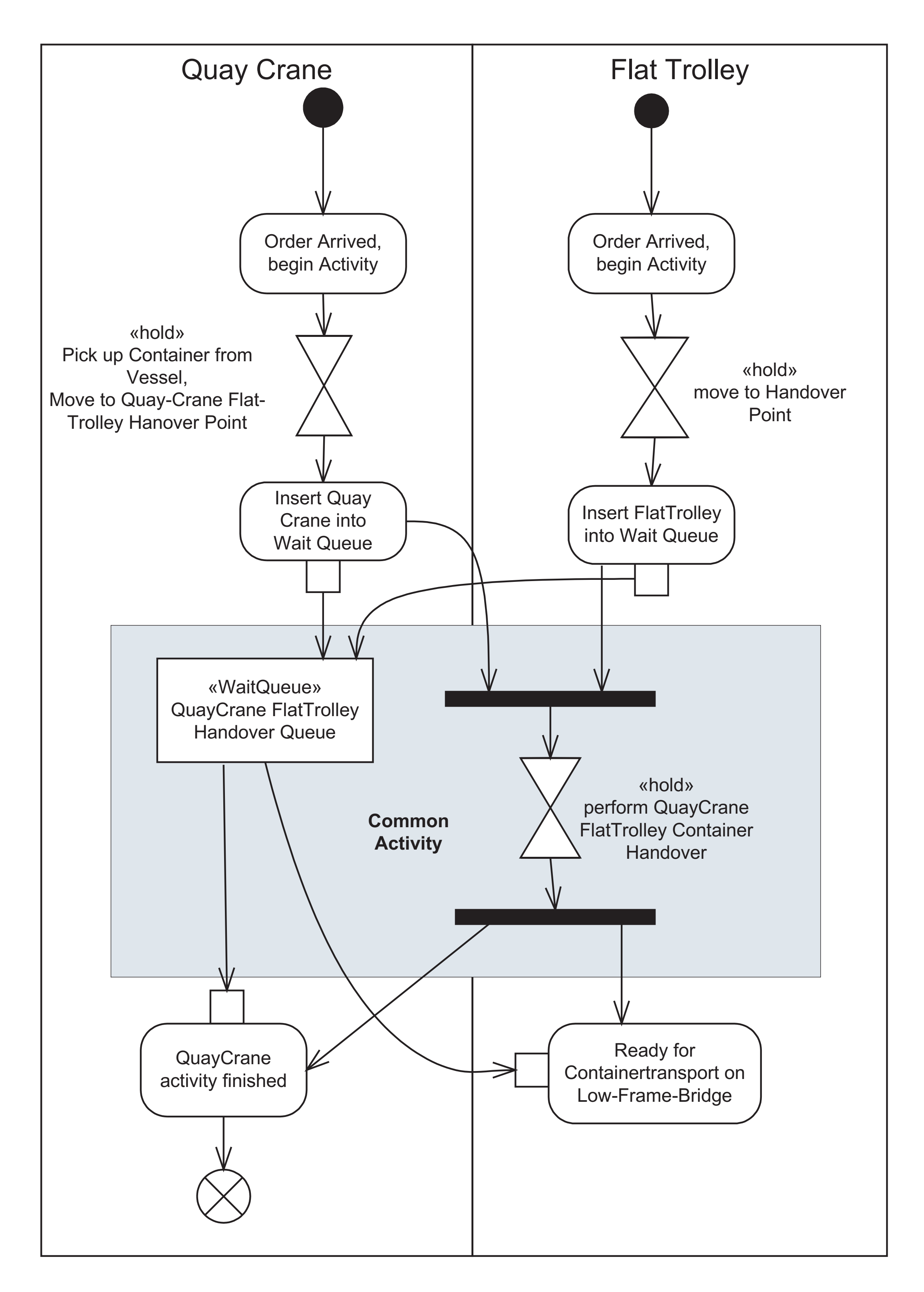


Abb. 5.1.: **Interaktion zwischen Containerbrücke und Flat-Trolley**

In einer weiteren Kooperation zwischen Lifting-Trolley und Terminal-Chassis wird der Container abgesenkt, gedreht und auf dem Chassis abgesetzt. Der Lifting-Trolley hebt seinen Spreader wieder auf volle Höhe und bearbeitet etwaige weitere Transportaufträge aus seiner internen Auftragswarteschlange bzw. deaktiviert sich und wartet auf weitere Aufträge.

Nach Ende seiner Interaktion mit dem Lifting-Trolley fährt das Terminal-Chassis den Container in den Stack auf die Höhe des Containerslots, in den der Container letztendlich eingelagert werden soll. Dort wartet es auf eine letzte synchronisierte Kooperation mit einem Portalkran, der den Container vom Chassis entnimmt, auf volle Höhe anhebt und ihn an der im Transportauftragsobjekt spezifizierten Stelle absetzt. Dann hebt er den Spreader auf volle Höhe und gibt sich und seinen Lagerblock als Ressourcen wieder frei.

Seeseitige Export-Aufträge beschreiben den Transport eines Containers aus dem Stack auf ein Containerschiff. Dabei sind die verwendeten Betriebsmittel dieselben wie beim seeseitigen Import, allerdings findet der Transport in umgekehrter Richtung und mit entgegengesetzter Reihenfolge der Ressourcenfreigabe statt.

Restacking-Aufträge

Containertransporte zur Organisation der Container innerhalb des Yards - die Restacking-Aufträge - umfassen entweder einen oder mehrere Containerblöcke. Innerhalb eines Containerblocks wird bei einem kleinen Abstand zwischen Ausgangs- und Zielcontainerslot nur ein Portalkran verwendet, der den Container anhebt, ihn zu seinem Zielpunkt fährt und dort ablegt. Größere Abstände werden mithilfe des Chassis überwunden, wobei der erste Portalkran den Container aus dem Block auf das Terminal-Chassis lädt, dieses den Container auf die Höhe der Einlagerposition fährt und dort vom zweiten Portalkran entladen wird, der den Container dann an der vorgesehenen Stelle in den Containerblock absetzt. Restacking-Aufträge zwischen zwei Containerblöcken werden durch einen Transport über die Low-Frame-Bridge umgesetzt. Das Vorgehen entspricht dabei dem Auslagern aus dem Block analog zum seeseitigen Export und dem Einlagern analog zum seeseitigen Import sowie einem Transport durch einen Flat-Trolley und dem Umpacken zwischen Chassis und Flat-Trolley mit einem Lifting-Trolley. Auch hier werden zunächst

alle Betriebsmittel mit Ausnahme des Flat-Trolleys reserviert und dann sukzessive wieder freigegeben.

Landseitige Transportaufträge

Landseitige Transportaufträge finden immer zwischen einem Stack und dem LKW-Gate an seiner Stirnseite statt. Dabei werden zwei Fälle unterschieden. Befindet sich der Containerslot in den ein- oder ausgelagert werden soll auf der landseitigen Hälfte des Stacks, so wird der Transportauftrag von einem Portalkran alleine durchgeführt. Im Falle eines Import-Auftrages fährt der hintere Portalkran über den LKW, hebt den Container an und fährt ihn selbständig an die Einlagerposition und senkt ihn dort ab. Bei einem Exportauftrag verfährt er in umgekehrter Reihenfolge. Befindet sich der Containerslot im seeseitigen Teil des Stacks, so wird der Containertransport von zwei Portalkranen durchgeführt, wobei sich einer über dem LKW und einer über der Position des Containerslots befindet. Der eine Kran entnimmt den Container und lädt ihn auf das Chassis des Stacks. Dieses fährt daraufhin zum anderen Portalkran, der den Container dann seinerseits je nach Auftragstyp entweder in den Stack einlagert oder auf dem LKW absetzt.

5.4. Aufbau des Systemzugangs

Alle zu verarbeitenden Transportaufträge werden vor Beginn der Simulation über eine grafische Benutzeroberfläche definiert. Sie liegen im Zeitpunkt der Simulationserstellung in Form eines Szenario-Objektes, dem `ParameterSingleton` vor. Vor Beginn eines Simulationslaufs generiert ein Objekt des Typs `InputToOrderConverter` alle im Szenario beschriebenen Tranportaufträge in Form der Modell-Entität des Typs `Order`. Dabei wird eine Parametrisierung vorgenommen, die den Auftrag umfassend beschreibt. Sie enthält die Art des Auftragstyps, um den es sich handelt, also ob ein land- oder seeseitiger Im- oder Export-Auftrag oder ein Restacking-Auftrag vorliegt. Des weiteren werden die Start- und Zielpunkte des Transportes gemäß des Szenarios festgelegt.

Für seeseitige Im- und Exportaufträge sind diese Punkte ein `QuayCraneProcess` und ein `StackingProcess`, die bei der Parametrisierung entweder explizit oder in Form einer

Liste angegeben werden, aus der bei der Auftragsgenerierung der angegebenen Prozesse zufällig ausgewählt wird. Für Restacking-Aufträge befinden sich Start- und Endpunkt des Containertransports innerhalb des Yards. Diese werden ebenso zufällig aus einer angegebenen Liste von Stacks zugewiesen.

Bei allen seeseitigen Aufträgen sowie den Restacking-Aufträgen, die zwei Stacks umfassen, findet ein Teil des Transportauftrags auf der Low-Frame-Bridge statt. In diesem Fall wird dem Auftrag der benötigte Bereich als `LaneReservation` übergeben. Landseitigen Transportaufträgen, also LKW Im- und Export, wird nur ein `StackingProcess` zugewiesen. Dies geschieht wieder entweder explizit oder aus einer Menge von möglichen Stacking-Prozessen. Soll ein `Order` einen Transportauftrag eines verspäteten LKW oder Containerschiffs repräsentieren, wird ihm dies noch als Eigenschaft zugewiesen.

Nachdem der Auftrag vollständig spezifiziert ist, wird ein Auftragsankunfts-Ereignis erstellt und mit dem Zeitpunkt der Auftragsankunft in die Ereignisliste des Schedulers eingefügt. Dieser Zeitpunkt wird als Teil des Szenario-Objekts für jeden Auftrag spezifiziert. Er definiert, ab wann ein Transportauftrag im System verfügbar ist. Das Eintreten des Ereignisses fügt das Auftrags-Objekt `Order` in die Warteschlange des Containerterminals ein. Von diesem Zeitpunkt an ist der Auftrag für das Simulationsmodell vorhanden und wartet auf seine Abarbeitung durch das System.

5.4.1. Steuerung des Systemeingangs

Die Entscheidung, ob ein Auftrag in das Containerterminal gelangt und damit der Containertransport von den beteiligten Betriebsmitteln durchgeführt wird, wird durch den für die Containerterminal-Steuerung zentralen `OrderAllocator`- Prozess getroffen. Dieser überprüft für einen Auftrag, ob alle Vorbedingungen erfüllt sind, so dass mit dem Containertransport begonnen werden kann. Welche Aufträge auf ihre momentane Durchführbarkeit überprüft werden, geschieht gemäß einer Regel, die bei der Parametrisierung des Szenarios angegeben werden muss (Abschnitt 4.4). Dabei stehen dem Anwender drei Auswahlmöglichkeiten zur Verfügung:

- *Strikte Ordnung:* Es wird nur für den ersten Transportauftrag der Containerterminal-Warteschlange untersucht, ob die Bedingungen zum Beginn des Containertransport

erfüllt sind.

- *Greedy:* Es wird die gesamte Warteschlange nach dem ersten Auftrag durchsucht, für den die Bedingungen zur Durchführung gegeben sind.

- *Strikte Ordnung an den Containerbrücken:* Zur Abbildung dieser Steuerungsregel wurde in jedem `QuayCraneProcess` eine Warteschlange eingefügt, über die eine feste Abarbeitungsreihenfolge erzwungen werden kann. Analog zur „greedy"-Auftragsallokation überprüft auch hier der Auftrags-Allokator alle Aufträge. Die Einhaltung der Reihenfolge an der Containerbrücke ist Teil der Auftragsvorbedingung.

Die Auftrags-Allokation findet immer dann statt, wenn entweder neue Aufträge in der Warteschlange ankommen oder wenn eine Ressource des Terminals freigegeben wird, sie also ihren Anteil an einem vorherigen Transport geleistet hat. Dazu wird der `OrderAllocatorProcess` entweder durch ein Auftragsankunfts-Ereignis oder durch den Simulationsprozess eines Betriebsmittels aktiviert, nachdem dieser sich als Ressource wieder freigegeben hat.

5.4.2. Prüfung der Systemeingangsbedingung

Die Aufträge, die nach der zuvor definierten Regel einen Containertransport beginnen könnten, werden vom Auftrags-Allokator auf eine `SystemEntranceCondition` geprüft, in der verschiedene je nach Auftragstyp unterschiedliche Teilbedingungen erfüllt sein müssen. Im Kern ist es erforderlich, dass alle für den Transport nötigen Ressourcen verfügbar sind.

Für alle Auftragstypen muss sichergestellt sein, dass der zugehörige `StackingProcess` und damit auch seine mit ihm verbundenen `Chassis`- und `GantryCraneProzesse` verfügbar sind. Für einen Restacking-Auftrag über zwei Containerblöcke müssen beide Stacking-Prozesse frei sein. Bei den seeseitigen Im- und Export-Aufträgen muss zusätzlich der beteiligte `QuayCraneProcess` verfügbar sein. Im Falle der Parametrisierung der Steuerungsregel als „Strikte Ordnung an den Containerbrücken" wird darüber hinaus geprüft, ob der Auftrag der erste in der Warteschlange an der Containerbrücke ist.

Handelt es sich beim Transportauftrag um einen seeseitigen Im- oder Export oder einen Restacking-Auftrag über zwei Lagerblöcke, muss zusätzlich die Verfügbarkeit eines Flat-Trolleys überprüft werden.

Während alle übrigen Modellkomponenten an höchstens einem Transportauftrag zur Zeit beteiligt sind, können auf der Low-Frame-Bridge mehrere Containertransporte gleichzeitig durchgeführt werden. Diese werden zum einen auf die verschiedenen Horizontalspuren verteilt, wobei kein weiterer Synchronisationsaufwand nötig ist, da die Spuren unabhängig von einander parallel befahrbar sind. Zum anderen können jedoch auch auf einer Spur mehrere Transportaufträge gleichzeitig durchgeführt werden, sofern diese unterschiedliche Bereiche derselben Horizontalspur verwenden. Um dabei Kollisionen zu vermeiden, besitzt das Modell eine zentrale Stelle zur Abstimmung der einzelnen `FlatTrolleyProcesse`, einen `FlatTrolleyAllocator`. Dieser hat Zugriff auf alle `FlatTrolleyProcesse`, ihren aktuellen Zustand und die von ihnen ausgehenden Reservierungen.

Dort werden alle `FlatTrolleyProcesse` bei ihrer Erzeugung angemeldet. Wird nun eine `SystemEntranceCondition` abgefragt, so prüft der `FlatTrolleyAllocator`, ob ein Flat-Trolley den Auftrag durchführen kann. Dies ist der Fall, wenn es eine Spur gibt, auf der ein freier Flat-Trolley[3] Start- und Endpunkt des Bereichs der Spur erreichen kann, über den der Container transportiert werden soll oder er ausschließlich durch andere Flat-Trolleys blockiert wird, die außerhalb des benötigten Bereichs genügend Platz zum ausweichen haben.

5.4.3. Freigabe eines Auftrages in das Containerterminal

Sind für einen überprüften Transportauftrag alle Vorbedingungen erfüllt, so erzeugt der `OrderAllocator` ein `SystemEntranceEvent`.

Durch dieses Ereignis werden dem betreffenden Auftrag die Containerslots im Stack zugewiesen, in die der Container ein- oder ausgelagert werden soll. Sie werden bei seeseitigen Export-Aufträgen gleichverteilt über die seeseitige Hälfte des Stacks generiert,

[3]Frei bedeutet, dass das Flat-Trolley keinem anderen Transportauftrag zugeordnet ist und nicht gerade den reservierten Bereich eines anderen Flat-Trolleys verlässt.

um ein Prestacking der Stellplatzplanung zu implizieren. Bei den übrigen Transportaufträgen werden sie über den gesamten Stack verteilt erzeugt. Der gewählte Containerslot wird dem Auftrag als gewählter Start bzw. Zielpunkt für den Containertransport übergeben. Bei LKW Im- und Exporten wird der Auftrag mit einer zufällig gewählten LKW-Ladebucht des Gates seines Stacks versehen.

Handelt es sich beim Transportauftrag um einen seeseitigen Im- oder Export-Auftrag oder um einen Restacking-Auftrag über zwei Containerblöcke, so muss ihm noch ein Flat-Trolley zugewiesen werden. Dazu wird am `FlatTrolleyAllocator` die Methode:

```
public FlatTrolleyProcess allocateFlatTrolley(Order order)
```

aufgerufen. Diese gibt den für den Auftrag günstigsten Flat-Trolley zurück.

Dabei wird folgende Priorisierung vorgenommen (Abb. 5.2):

1. Gibt es eine Horizontalspur mit einem freien Flat-Trolley, der sowohl den Start- als auch den Endpunkt des für den Transport nötigen Bereichs erreichen kann, dass bedeutet es befinden sich keine weiteren Flat-Trolleys in diesem Bereich und kein Abschnitt des Bereichs überschneidet sich mit der Reservierung eines anderen Flat-Trolleys, so wird dieser ausgewählt.

2. Trifft Punkt eins auf mehrere Spuren und Flat-Trolleys zu, so wird derjenige gewählt, der dem Startpunkt des Bereichs am nächsten ist.

3. Ist auf keiner Spur der Weg für einen freien Flat-Trolley vorhanden, um Start- und Endpunkt des Bereichs zu erreichen, so wird derjenige gewählt, der ausschließlich von freien Flat-Trolleys blockiert wird, sofern für diese außerhalb des Bereichs genügend Raum zum Ausweichen ist.

4. Trifft Punkt drei auf mehrere Flat-Trolleys zu, so wird derjenige gewählt, der dem Startpunkt am nächsten ist.

In einem weiteren Schritt übergibt das `SystemEntranceEvent` allen nun im Transportauftrag spezifizierten Betriebsmitteln den Auftrag als ihren aktuellen Auftrag, setzt ihren Zustand auf beschäftigt und aktiviert sie.

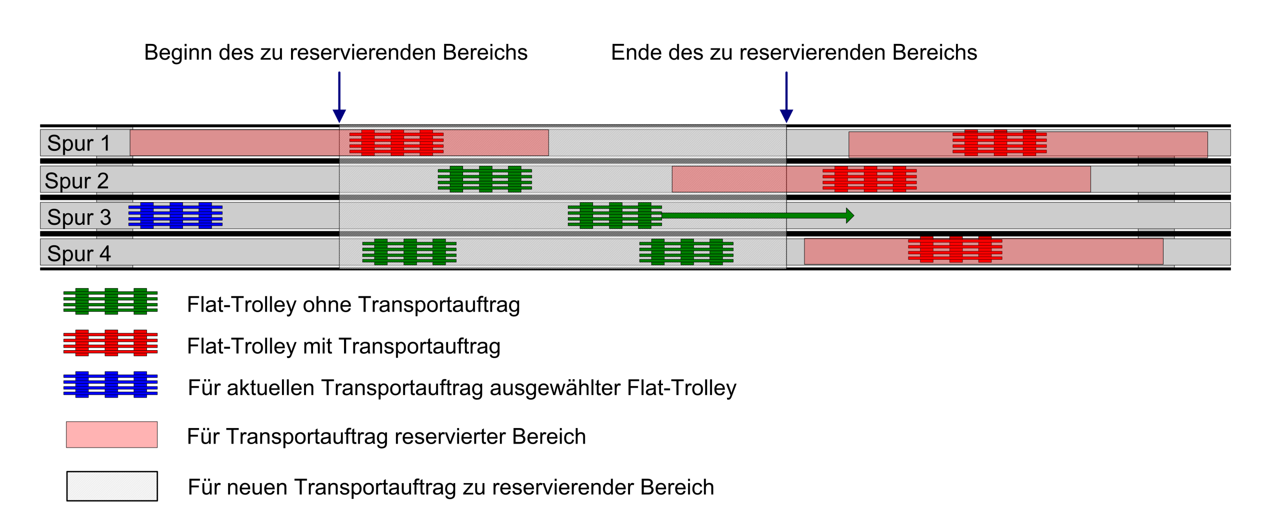


Abb. 5.2.: **Zuordnung eines Flat-Trolleys zu einem Transportauftrag**

5.5. Abbildung stochastischen Modellverhaltens

Eins der Hauptanwendungsfelder von Simulation ist die Untersuchung von Systemen, in denen Teile ein nicht-deterministisches Verhalten besitzen. Diese beschreiben zufällige Ankunftszeiten an den Systemaußengrenzen, Verspätungen, Bedienzeiten, menschliches Verhalten und Geräteausfälle. Dazu werden die Zufallsprozesse des Realsystems auf Wahrscheinlichkeitsverteilungen abgebildet, die diese möglichst präzise beschreiben. Durch die Verwendung von Quasi-Zufallszahlen dieser Verteilungen wird in einem Experiment ein mögliches Verhalten des simulierten Systems nachempfunden. Mit einer Reihe von Simulationsläufen lässt sich so ein sehr viel realistischeres Bild des wirklichen Systems erreichen, als dies mit festen, gemittelten Werten in einem einzelnen deterministisch berechenbaren Simulationslauf möglich wäre. Das Erkennen, wo im abgebildeten System Zufallsprozesse stattfinden und diese präzise durch stochastische Verteilungen zu beschreiben, ist also eine der zentralen Aufgaben bei der Abbildung auf ein Simulationsmodell. Je besser die stochastischen Verteilungen das Ausgangssystem beschreiben, umso realistischer ist das Simulationsmodell und umso verlässlicherer sind die Aussagen, die sich aus den Simualtionsergebnissen auf das Realsystem ableiten lassen. ([PK05], S. 31 und Kap. 7)

Für die Untersuchung des Horizontalkrankonzepts wird ein Containerterminal mit den neuartigen Betriebsmitteln modelliert, um generelle Aussagen hinsichtlich bestimmter

Kenngrößen im ordentlichen Betrieb treffen zu können. Deshalb wird von Maschinenausfällen bewusst abstrahiert. Die Abbildung menschlichen Verhaltens ist insofern nicht sinnvoll, als dass die Steuerung der Geräte weitgehend automatisiert stattfindet und große Abweichungen von der Norm nicht zu erwarten sind. Ebenso sind die Bedien- bzw. Transportzeiten der einzelnen Terminal-Betriebsmittel über ihre Geschwindigkeiten explizit modelliert, so dass sich Abweichungen aus dem Modell heraus erklären und nicht auf stochastische Prozesse abgebildet werden müssen.

Die im Modell nicht explizit über ein konkretes deterministisches Verhalten abgebildeten Bereiche des Systems Containerterminal sind die Stau- und die Stellplatzplanung (Abschnitt 4.4) und mögliche Verspätungen von Containerschiffen und LKW.

Containerschiff- und LKW-Ankunftszeiten und -verspätungen

Die Schiffs- und LKW-Ankünfte an einem Containerterminal sind in der Regel im Voraus bekannt. Deswegen werden sie in Form von Ankunftszeitpunkten dem Modell übergeben. Mit diesen festen Ankunftszeiten lässt sich das Containerterminal in seinem ordentlichen Betrieb simulieren. Da Verspätungen in einem realen Containerterminal vorkommen, kann ihre Abbildung den Grad an Realismus des Modells erhöhen. Deshalb soll es auch im vorliegenden Simulationsmodell möglich sein, Verspätungen abzubilden.

In der diskreten Ereignissimulation werden Verspätungen typischerweise entweder über eine stochastische Verteilung von Ankunftszeitpunkten oder mit einer Wahrscheinlichkeit für das verspätete Eintreffen und einer Verteilung für den Grad der Verspätung abgebildet. Die vorliegende Modellierung ist stark auf die Gewinnung von Kennzahlen für Teilprozesse des Systems wie die Ladezeit eines Schiffes ausgerichtet und bildet die Containerorganisation im Yard nicht explizit ab. Deswegen würde die Abbildung von Verspätungen auf eine veränderte Ankunftszeit im Modell die tatsächlichen Wirkungen nicht realistisch widerspiegeln. Die relative Dauer für das Beladen eines Schiffes bliebe unverändert und nachteilige Wirkungen auf den Vorstau würden nicht berücksichtigt.

Aus diesen Gründen werden Verspätungen im vorliegenden Simuationsmodell nicht durch eine veränderte Ankunftszeit sondern durch ihre Wirkung auf die Stellplatzplanung beschrieben. Es wird die Annahme getroffen, dass eine verspätete Schiffs- oder LKW-

Ankunft den Grad des Vorstaus verschlechtert, der zur Vorbereitung der jeweiligen Ankunft getroffen wird. Wenn ein Schiff rechtzeitig am Containerterminal ankommt, wird davon ausgegangen, dass die Stellplatzplanung alle Container für dieses Schiff so gestapelt hat, dass diese ohne weiteres Restacking aus den Containerblöcken entnommen und auf das Schiff gebracht werden können. Gleiches gilt für einen Container, der auf einen pünktlich ankommenden LKW verladen wird. Im Falle einer verspäteten Ankunft wird davon ausgegangen, dass der Vorstau bereits für andere Transportaufträge vorgenommen wurde und deswegen die Container nicht mehr optimal für die verspäteten LKW und Schiffe gestapelt sind. Dies wird durch eine pauschale Zeitspanne für die Mehrarbeit der Portalkrane für die Export-Aufträge ausgedrückt.

Dem Anwender wird dazu eine Parametrisierung der Verspätungen (Abschnitt 6.1.5) ermöglicht, bei der er für LKW und Schiffe jeweils die Wahrscheinlichkeit der Verspätung und die Verlängerung der Bearbeitungszeit für jeden verspäteten Export-Auftrag angeben kann. Für die Dauer der Mehrarbeit ist in der Standard-Parametrisierung bereits der Wert 125 s angegeben. Dieser entspricht einem einfachen Restacking-Auftrag durch einen Portalkran zwischen zwei einander nahe gelegenen Containerslots.

Initialisierung der Wahrscheinlichkeitsverteilungen

Im vorliegenden Simualtionsmodell wurden an drei Stellen Zufallsprozesse abgebildet. Sie dienen der Zuordnung von Containerbrücken und Stacks zu einer Gruppe von Transportaufträgen, der Abbildung zufälliger Containerslots auf einen Transportauftrag und der Zuordnung von Verspätungen zu LKW und Containerschiffen.

Zur Abbildung der einzelnen Zufallsprozesse werden Wahrscheinlichkeitsverteilungen verwendet. Dabei lassen sich alle drei mit Zufallswerten aus einer Gleichverteilung beschreiben. Um dem Anwender die Möglichkeit zu geben, zu untersuchen, welchen Einfluss einer der abgebildeten Zufallsprozesse auf das Systemverhalten hat, wird jeder der Prozesse über eine eigene Verteilung modelliert. Sie liefert einen von den anderen unabhängigen Strang an Zufallszahlen ([PK05], S. 161 ff.). Dazu ist es in der Parametrisierung des Modells möglich, jede dieser Verteilungen mit einem eigenen Seedwert zu initialisieren.

Eine mit einem Seedwert initialisierte Verteilung erzeugt immer denselben Strang von Pseudozufallszahlen, so dass dieser reproduzierbar wird. Durch die Veränderung eines Seedwertes über mehrere Simulationsläufe bei einem sonst gleichen Szenario mit gleichen Seedwerten für die übrigen Verteilungen kann der Anwender den Einfluss einer der Verteilungen auf das Verhalten des Modells untersuchen und damit Rückschlüsse auf den Einfluss auf das abgebildete System ziehen.

6. Ein- und Ausgaben des Modells

Die umfassende Untersuchung des Horizontalkrankonzepts mithilfe eines Simulationsmodells erfordert von diesem weitreichende Möglichkeiten der Parametrisierung, um verschiedene Zusammenhänge des Konzepts in ihrem Einfluss auf das Systemverhalten zu bestimmen. Dazu wird eine grafische Benutzerschnittstelle geschaffen, die es dem Anwender ermöglicht Modelleingaben strukturiert in einer Maske vorzunehmen, auf Default-Werte zurückzugreifen und Szenarien zu speichern, zu laden und zu verändern. Die Ausgaben des Modells untergliedern sich in eine Animationsdatei, in der das Modellverhalten nach einem Simulationslauf grafisch nachvollzogen werden kann und eine Ausgabe im Microsoft Excel-Format. Diese enthält strukturiert aufgeführt die Parametrisierung eines Szenarios sowie eine Vielzahl statistischer Kennzahlen zur Auswertung bezüglich der verschiedenen Auftragstypen, Schiffe, Schiffstypen und zusammengefasster Werte.

6.1. Parametereingabe

Zur Analyse des Horizontalkrankonzeptes ist es sinnvoll diese Architekturvariante mit einer Reihe verschiedener Parametrisierungen zu untersuchen. Dazu werden verschiedene Einstellungsmöglichkeiten bezüglich des Modells, der Auftragsankünfte am Modell und des einzelnen Experiments umgesetzt. Um eine leichte Eingabe der Parameter zu ermöglichen, wird eine grafische Benutzeroberfläche erstellt.

Bei der Wahl, mit welchen Mitteln die Parametereingabe des Modells grafisch realisiert werden soll, werden zunächst zwei unterschiedliche Alternativen untersucht. Die eine bildet die Verwendung des Eclipse-Plugins Simtools, die andere die eigenständige Entwicklung einer grafischen Benutzeroberfläche. SimTools wurde am Arbeitsbereich „An-

gewandte und sozialorientierte Informatik“ des Departments Informatik der Universität Hamburg entwickelt.

Sim Tools ist ein Desmo-J unabhängiges Werkzeug zur Parametrisierung von Simulationsmodellen. In ihm sind Optimierungsmethoden und eine grafische Auswertung der Ergebnisse integriert. Für die Verwendung von SimTool spricht, dass in diesem Fall ein weitaus geringerer Aufwand zu betreiben ist, als bei der eigenständigen Programmierung einer Benutzerschnittstelle. Das Vorliegen in Form eines Eclipse-Plugins erscheint in sofern günstig, als dass Eclipse als Umgebung für die Programmierung des Modells verwendet wird. Bei der Verwendung von SimTools muss für das Modell eine Wrapper-Klasse geschrieben werden. In dieser müssen alle Werte des Modells, die auf veränderbare Parameter abgebildet werden, spezifiziert werden. Für die Parametrisierung wird vor Beginn eines Experiments eine Eingabemaske geöffnet, in die die Parameterwerte für ein bestimmtes Szenario eingegeben werden. Im Rahmen einer Simulationsoptimierung können für einen Parameter auch verschiedene Werte eingegeben werden, die dann über mehrere Simualtionsläufe automatisch verwendet werden.

Gegen die Verwendung spricht, dass SimTools nur die Eingabe primitiver Datentypen zulässt. Beim vorliegenden Modell sind jedoch einige Parameterwerte von einander abhängig, so dass eine Plausibilitätsprüfung bei der Eingabe wünschenswert ist. Einige Eingabeparameter des Containerterminal-Modells sind auf bestimmte unterschiedliche Werte festgelegt, so dass dem Benutzer eine Fülle von Zusatzinformationen zur Benutzung des Modells mitgegeben werden müsste. Dass es sich bei SimTools um ein Eclipse-Plugin handelt, erleichtert zwar die Programmierung des Modells, da es keine Brüche zwischen der Umgebung zur Erstellung des Modells und der Visualisierung der Eingaben gäbe, allerdings ist die grafische Parametereingabe ohne Eclipse nicht möglich. Ein weiterer Nachteil von SimTools ist die fehlende Dokumentation, die eine Verwendung erschweren würde.

Letztendlich wird davon Abstand genommen SimTools zu verwenden. Es wird eine eigenständige Benutzeroberfläche aufgebaut.

Modellierung der Benutzerschnittstelle mit Netbeans IDE

Die Wahl des Entwicklungswerkzeuges fällt dabei auf Netbeans IDE[1]. Netbeans IDE ist eine umfangreiche Programmierumgebung, die auch Java unterstützt. Da für die Erstellung des Simulationsmodells in weiten Teilen die Entwicklungsumgebung Eclipse SDK[2] verwendet wird, stellt es zwar einen gewissen Aufwand dar, das Programmierprojekt in zwei unterschiedlichen Programmierumgebungen durchzuführen. Netbeans erlaubt es jedoch auf einem bestehenden Eclipse-Workspace zu arbeiten. So kann die Entwicklung recht leicht synchron durchgeführt werden. Im Gegensatz zu Eclipse bietet Netbeans die Möglichkeit grafische Elemente wie Textfelder, Schaltflächen und Auswahlboxen per Drag and Drop zu einer Benutzeroberfläche zusammenzufügen. Dabei können sowohl Swing[3] als auch AWT[4] Komponententen verwendet werden. An dieser können bei der grafischen Modellierung schon verschiedene Einstellungen vorgenommen werden, aus denen automatisch der entsprechende Programmcode generiert wird. Dieser liegt in Form einer Java-Klasse des Typs JFrame[5] vor und kann nach den Anforderungen der Anwendung erweitert werden. Dabei kann der Entwickler frei zwischen dem Entwurf der Programmoberfläche und dem Quellcode der Klasse wechseln.

6.1.1. Struktur der grafischen Benutzerschnittstelle

Bei der Parametrisierung eines spezifischen Simulationslaufs lassen sich unterschiedliche Arten von Parametern unterscheiden, je nachdem für welchen Bereich des Simulationslaufs sie eine Eingabe erlauben (Abschnitt 4.1.3). Es gibt Einstellungsmöglichkeiten für die Organisation der Input- und Output-Dateien, Variationen der Experinentdauer, der Dimensionerung des Modellaufbaus und der technischen Daten seiner Betriebsmittel, der Steuerstrategie, der Schiffs- und Auftragsankünfte eines Szenarios sowie der Realisierung stochastischen Verhaltens. Zur Erhöhung der Übersichtlichkeit und des Verständnisses,

[1]NetBeans IDE, Version 6.5
[2]Eclipse SDK, Version 3.4.1
[3]Programmierschnittstelle und Grafikbibliothek zum Programmieren grafischer Benutzeroberflächen
[4]Standard-API zur Erzeugung und Darstellung einer plattformunabhängigen grafischen Benutzerschnittstelle für Java-Programme, älter und mit einem geringeren Funktionsumfang als Swing
[5]javax.swing.JFrame

wofür eine bestimmte Eingabe sorgt, werden auf der Benutzeroberfläche die Eingabemöglichkeiten nach Art der Parameter gruppiert und zu einer Registerkarte zusammengefasst (Abb. 6.1).

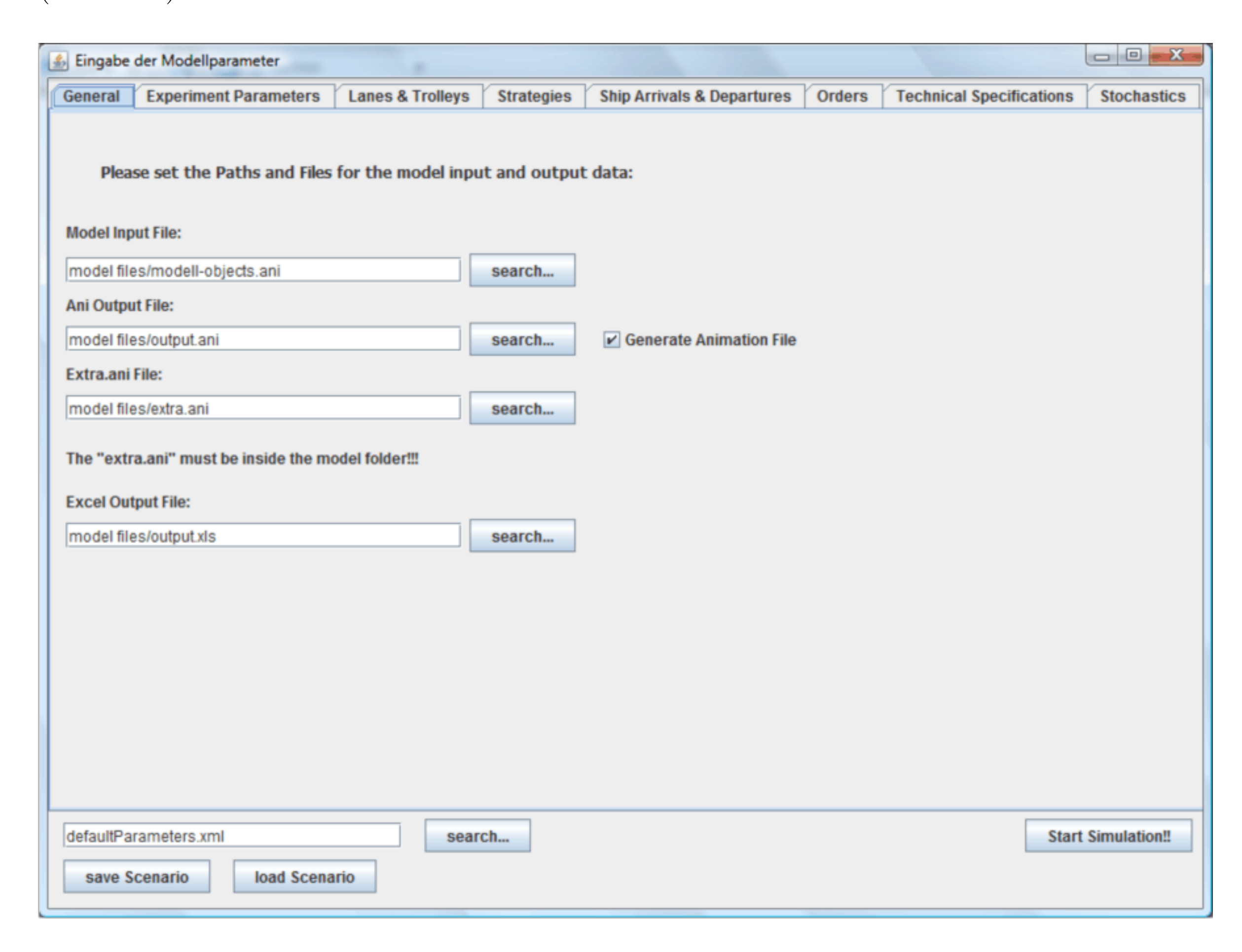


Abb. 6.1.: **Parametereingabe über die Benutzerschnittstelle**

6.1.2. Interne Repräsentation des Eingabeobjetes

Alle Eingaben der Parametrisierung werden intern auf ein einzelnes Objekt der Klasse `ParameterSingleton` abgebildet. Dort werden sie in Form eines jDom-Dokumentes (Abschnitt 4.6) vorgehalten. Diese Modellierung ermöglicht eine Hierachisierung der Eingaben bezüglich der Art und Abhängigkeiten der Parameter in der Baumstruktur des jDom-Objektes. Zu Beginn der Simulation wird dieses Objekt als die Gesamtheit aller vorgenommenen Parameter-Einstellungen dem Simulationsmodell übergeben. Die so

vorliegenden Werte können aus dem jDom-Objekt als XML-Dokument gespeichert und geladen werden. Dabei entspricht eine gespeicherte XML-Datei (Abschnitt A.2) einer genauen Repräsentation des Datenobjektes der Parametereingabe, so dass die gespeicherte Szenariodatei aus sich heraus gut verständlich ist und in einem XML-Editor manipuliert werden kann. Ein Nachteil dieser Variante ist allerdings, dass alle Werte in Form von Zeichenketten (Strings) übergeben werden müssen, so dass bei dieser Modellierung ein Zusatzaufwand durch Typumwandlungen entsteht. Zu Beginn der Eingabe werden alle Kennziffern des Modells mit Standardparametern belegt. Sie ersparen dem Benutzer des Modells die Belegung von Werten, die er übernehmen kann, insbesondere die der Modellpfade und technischer Parameter. Um mehrere ähnliche Parametrisierungen zu untersuchen, speichert das Programm die getroffenen Einstellungen automatisch als „letzten Simulationslauf“. Dieser kann geladen und dann entsprechend modifiziert werden.

6.1.3. Parametrisierung der Modelldateien

Bei der Eingabe der Modellparameter sind auf der ersten Registerkarte die Input- und Output-Dateien mit ihren Pfaden zu hinterlegen (Abschnitt A.2, Zeile 18-25). Dort muss zunächst die Modelldatei spezifiziert werden, in der sich das in Verdigo erstellte Layout befindet. Des weiteren müssen Pfad und Name der Animationsdatei angegeben werden, die nach dem Simulationsexperiment eine Darstellung in Golem 3D erlaubt. Da die Erzeugung einer Animationsdatei einen recht hohen Ressourcenaufwand bedeutet, sowohl bezüglich der Rechenzeit des Simulationsexperimentes als auch der Größe der Animationsdatei, kann darauf auch verzichtet werden, indem der Haken aus der Checkbox „Generate Animation File“ entfernt wird. Des Weiteren wird noch die „extra.ani“ übergeben, in der die Zeitpunkte angegeben sind, an denen sich die Sichtbarkeit der 3D-Modell-Komponenten der Containerschiffe und -brücken verändert sowie angegeben ist, welche Komponenten der Low-Frame-Bridge Teil des aktuellen Szenarios sind. Sie sind für die Durchführung der Animation essentiell. Zuletzt muss noch die Datei für die Excel-Ausgabe spezifiziert werden.

Um dem Nutzer die Möglichkeit zu geben, bestehende Dateien und Dateipfade auszuwählen, ohne die Eingabe per Hand vornehmen zu müssen, wird für jedes der Eingabefelder

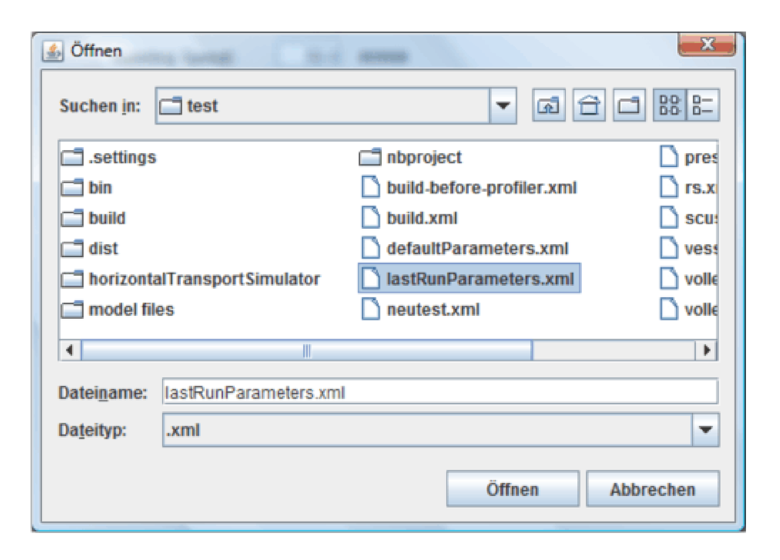


Abb. 6.2.: **Dateiauswahl über ein Dialogfenster**

ein Dateiauswahl-Dialog angelegt. Das Simulationsmodell soll mit den ihm anhängigen Dateien zwischen unterschiedlichen Systemen portierbar sein. Deshalb werden alle Dateipfade nach der Dateiauswahl zu relativen Pfaden konvertiert.

6.1.4. Parametrisierung des Experiments

Die Länge eines Simulationslaufes in Desmo-J lässt sich über die Angabe des Start- und des Endzeitpunktes eines Experimentes spezifizieren. Außerdem ist es möglich einen Trace zu erzeugen, der nach Zeitpunkten geordnet alle Ereignisse des Schedulers enthält. Dazu gehören die Aktivierung und Passivierung von Prozessen, die Ausführung von Kooperationsobjekten und alle Ereignisse. Eine weitere Ausgabe, die für ein Experiment definierbar ist, stellt die Debug-Periode dar. Diese Zeitabschnitte sind jeweils mit ihren Start- und Endzeitpunkten spezifizierbar. Für ihre Parametrisierung wurde jeweils ein formatiertes Textfeld bereitgestellt, das ausschließlich die Eingabe rationaler Zahlen erlaubt, um so Fehleingaben im Vorwege auszuschließen. Dieses wurde nicht nur für das Experiment, sondern für alle Eingaben von Zeitpunkten in Felder des Benutzerinterfaces so realisiert (Abschnitt A.2, Zeile 3-17).

6.1.5. Parametrisierung von Modelldaten

Die Eingabe von Modellparametern untergliedert sich in den Aufbau der Low-Frame-Bridge, die Auswahl einer Steuerung für den Modelleintritt, die technische Spezifikation der eingesetzten Geräte und die Belegung von Werten für stochastische Prozesse. Für alle vier Kategorien wurde jeweils eine Registerkarte angelegt.

Die Parametrisierung der Dimensionen der Low-Frame-Bridge erlaubt es, die Anzahl von Horizontal-Spuren festzulegen. Dabei können zwischen einer und vier Spuren in einem Szenario am Containertransport beteiligt sein, die jeweils einzeln über eine Checkbox auswählbar sind. Spuren, die nicht Teil des aktuell zu untersuchenden Modells sind, werden in der Animation nicht dargestellt. Für jede Spur, die Teil der Parametrisierung des Modells ist, besteht die Möglichkeit über eine Combobox zwischen einem und vier Flat-Trolleys und einem und acht Lifting-Trolleys auszuwählen (Abschnitt A.2, Zeile 26-47). Ist der Haken in der Checkbox zu einer Spur nicht gesetzt, ist der Bereich zur Auswahl der Flat- und Lifting-Trolleys grau hinterlegt. Darüber und über die Auswahl der Anzahl der Flat- und Lifting-Trolleys aus Checkboxen wird erreicht, dass der Benutzer zum einen keine unzulässigen Eingaben vornehmen kann und zum anderen, dass er den Bereich kennt, aus dem er seine Auswahl zu treffen hat, womit eine höhere Benutzungssicherheit gegenüber einfachen Textfeldern erreicht wird.

Für die Eingangssteuerung des Modells wurden drei unterschiedliche Strategien hinterlegt (Abschnitt 5.4.1). Die Auswahl trifft der Benutzer über einen Radiobutton, so dass sichergestellt ist, dass genau eine dieser sich ausschließenden Regeln verwendet wird (Abschnitt A.2, Zeile 4).

Für die Parametrisierung der eingesetzten Geräte bezüglich ihrer technischen Spezifikationen werden formatierte Textfelder eingesetzt (Abschnitt A.2, Zeile 58-83). Dort können für die Containerbrücken, Portalkrane, Terminal-Chassis sowie Flat- und Lifting-Trolleys Einstellungen vorgenommen werden. Sie beschreiben die Fahrgeschwindigkeiten, Geschwindigkeiten für das Bewegen der Trolleys sowie für das Heben und Senken der Spreader. Des Weiteren wird hier die Dauer für das Aufnehmen und Absetzen eines Containers spezifiziert. Während modellintern alle Geschwindigkeiten in Meter pro Sekunde gespeichert werden, werden hier, um die Parametereingabe zu erleichtern und eine

bessere Lesbarkeit zu gewährleisten, für die Geräte üblichere Angaben in Kilometer pro Stunde bzw. Meter pro Minute verwendet.

Die Einstellungen, die in der Registerkarte „stochastik“ vorgenommen werden können, erlauben zum einen das Setzen von Seedwerten für die verwendeten stochastischen Verteilungen und zum die Abbildung von Verspätungen von Schiffen und LKW im Modell (Abschnitt A.2, Zeile 84-97). Es lassen sich für die zufällige Verteilung der in einem Auftrag zugewiesenen Containerslots, für die Auswahl von Containerbrücken und Stacks aus einer Liste bei der Auftragsgenerierung und für die Zufallszahlengenerierung für Verspätungen jeweils eigene Seedwerte zuweisen. Diese werden in ein für die Eingabe von natürlichen Zahlen formatiertes Textfeld eingegeben. Es ist außerdem möglich, alle Seedwerte über das Setzen eines Hakens in einer Checkbox automatisch erzeugen zu lassen. Bei den Verspätungen von LKW und Schiffen lässt sich jeweils die Wahrscheinlichkeit einer Verspätung und die Mehrarbeit angeben, die diese pro Export-Auftrag bedeuten.

6.1.6. Parametrisierung von Schiffs- und Auftragsankünften

Bei den bisher diskutierten Eingaben bestehen nur schwache Abhängigkeiten zwischen den einzelnen Werten. Deswegen müssen dort nur wenige Validitätsprüfungen vorgenommen werden. Diese beziehen sich meist auf das Format einer Eingabe oder darauf, ob überhaupt eine Eingabe zulässig ist. Bei der Parametrisierung der Schiffs- und Auftragsankünfte liegen weitaus größere Interdependenzen zwischen einigen der Parameter vor, so dass hier ein höherer Aufwand betrieben werden muss, um inkonsistente Eingaben zu verhindern. Dabei ist insbesondere darauf zu achten, dass immer nur ein Schiff zur Zeit einen Liegeplatz besetzen kann und seeseitige Aufträge nur dann auftreten können, wenn sie Bezug zu einem Schiff haben. Da sich dieses nicht innerhalb eines Transportauftrages oder einer Schiffsankunft bestimmen lässt, wird ein zentrales Objekt des Typs `QuayLineSpaceResolver` erzeugt, das die Konsistenzprüfungen vornimmt.

6.1.7. Schiffsliegezeiten

Die Registerkarte für die Parametereingabe der Schiffsliegezeiten ist zweigeteilt. Im oberen Bereich befindet sich eine Eingabemaske für die Spezifikation eines Schiffes, im un-

teren eine Tabelle aller bisher angelegten Containerschiffe (Abb. 6.3).

Um die Liegezeit eines Schiffs zu spezifizieren, muss einer der drei Liegeplätze des Terminalmodells angegeben und ein Schiffstyp ausgewählt werden. Der Schiffstyp ist aus fünf unterschiedlich großen Containerschiffen wählbar. Davon sind zwei Feeder-Schiffe. Mit der Wahl eines Schiffstyps wird im Auswahlfeld „Number of Quay Cranes“ ein Standardwert für diesen Schiffstyp gesetzt und die Ladedauer des Referenzmodells in das Feld „estimated Time of Loading“ eingefügt. Die Ankunftszeit und die nicht-produktive Liegezeit müssen für das Containerschiff angegeben werden, die erwartete Ladezeit kann gegenüber der Vorgabe aus dem Referenzmodell und die Anzahl der Containerbrücken gegenüber dem Standardwert verändert werden. Nachdem alle Werte eingegeben wurden, kann das Schiff über die „Add“-Schaltfläche in die Schiffstabelle und damit die Modellparametrisierung aufgenommen werden. Dabei überprüft zunächst der `QuayLineSpaceResolver`, der intern alle schon in der Tabelle befindlichen Schiffe enthält, ob sich während der angegebenen Liegezeit schon ein anderes Schiff an diesem Liegeplatz befindet. Ist das der Fall, wird die Schiffsankunft nicht in die Tabelle übernommen und es wird eine Warnung ausgegeben. Ist der Liegeplatz für den angegebenen Zeitraum frei, wird es in die Tabelle der Schiffsliegezeiten eingefügt. Der `QuayLineSpaceResolver` fügt das Schiff zusätzlich seiner internen Liste aller Schiffe hinzu.

Mit Betätigen der „Remove“-Schaltfläche kann ein Containerschiff aus der Tabelle wieder entfernt werden. Dazu prüft der `QuayLineSpaceResolver` zunächst, ob seeseitige Import- oder Exportaufträge an das Schiff gebunden sind. Ist dies der Fall, kann es nicht aus der Tabelle entfernt werden. Es wird eine diesbezügliche Warnung ausgegeben. Sind keine Aufträge an das Schiff gebunden, wird es aus der Liste und aus dem `QuayLineSpaceResolver`-Objekt entfernt.

6.1.8. Auftragsankünfte

Analog zur Registerkarte der Schiffsankünfte werden die Transportaufträge auch zunächst in einer Maske definiert und dann in eine Tabelle eingefügt (Abb. 6.4). Sie werden dabei immer als eine Gruppe von Aufträgen angelegt.

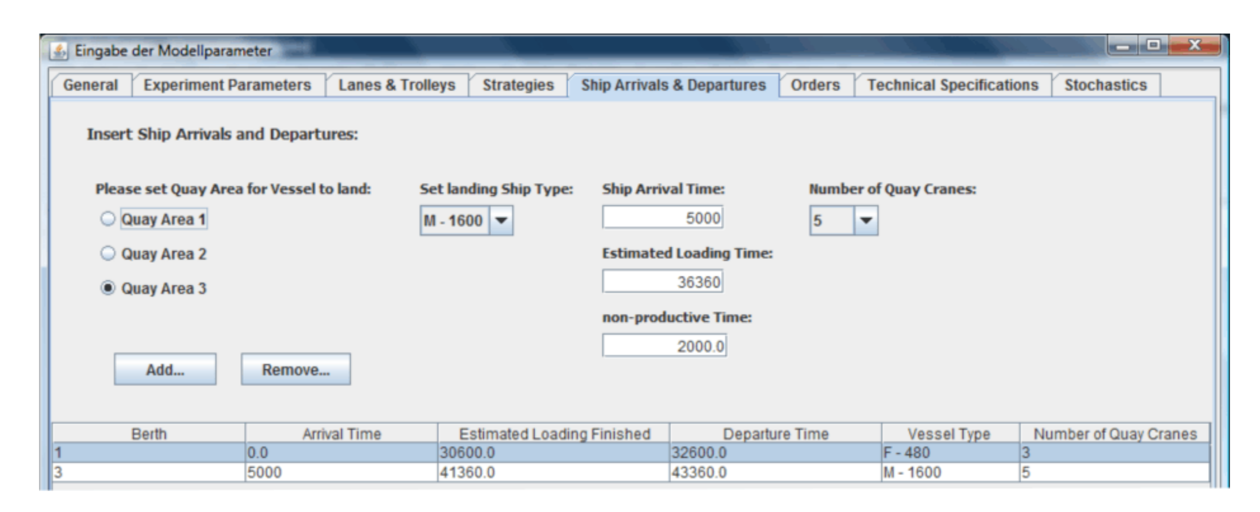


Abb. 6.3.: **Parametrisierung der Schiffsliegezeiten**

Um eine Gruppe landseitiger Transportaufträge zu definieren, muss bei der Wahl des Auftragstyps entweder „Import(land)“, „Export(land)“ oder „Import/Export(land)“ aus der Auftragstypen-Checkbox ausgewählt werden. Für „Import/Export“ muss zusätzlich über einen Schieberegler angegeben werden, welchen Anteil der jeweilige Typ an der Gesamtgruppe hat. Über Feldeingaben wird dann spezifiziert, wie viele Aufträge erzeugt werden sollen, wann der erste am Terminal ankommen und ein wie großer Zeitabstand zwischen den einzelnen Auftragsankünften liegen soll. Über Checkboxen werden zu den Aufträgen die beteiligten Stacks ausgewählt. Sind mehrere Stacks angegeben, finden die Aufträge gleichverteilt zufällig an einem der Stacks statt. Über die „Add“-Schaltfläche wird die angelegte Gruppe in die Auftragsliste übernommen. Restacking-Aufträge werden analog zu landseitigen Transporten angelegt. Der Unterschied besteht darin, dass dort die Anfangs- und End-Containerslots über alle angegenbenen Stacks verteilt werden.

Seeseitige Im- und Exportaufträge werden intern an ein Schiff gebunden (Abschnitt A.2, Zeile 53-54), das vorher auf der „Ship Arrival and Departures“-Registerkarte angelegt werden muss. Bei der Auftragsanlage müssen Liegeplatz, Auftragszeitpunkt, -Anzahl und -Typ sowie Zwischenabstände angegeben werden. Der `QuayLineSpace Resolver` überprüft, ob der Auftragsankunfts-Zeitpunkt innerhalb der Liegezeit eines Schiffs liegt und die Liegeplätze übereinstimmen. Ist das der Fall, sind die Containerbrücken des Schiffs aus einer Checkbox auswählbar. Nachdem die Stacks und die Containerbrücken ausgewählt wurden, kann die Auftragsgruppe der Liste hinzugefügt werden und wird

dabei intern im `QuayLineSpaceResolver` an das Schiff gebunden. In Analogie zur Definition landseitiger Transportaufträge können auch seeseitig Gruppen angelegt werden, die sowohl Im- als auch Exportaufträge enthalten. Auch hier werden über den Schieberegler die Mengenverhältnisse bestimmt. Darüber hinaus muss noch angegeben werden, wie lang ein „Auftragszyklus“ sein soll. Das heißt, wie viele Import-Aufträge von einer Containerbrücke abgearbeitet werden müssen, bevor diese mit der Durchführung von Exportaufträgen beginnt und umgekehrt. Damit wird abgebildet, dass ein bestimmter Bereich eines Schiffes, zum Beispiel eine Luke oder ein einzelner Containerstapel, erst komplett ausgeladen werden muss, bevor dort neue Container eingestellt werden können.

Bestehende Werte der Auftragsliste können über die Schaltflächen „Remove“ wieder entfernt und „Change Values“ verändert werden.

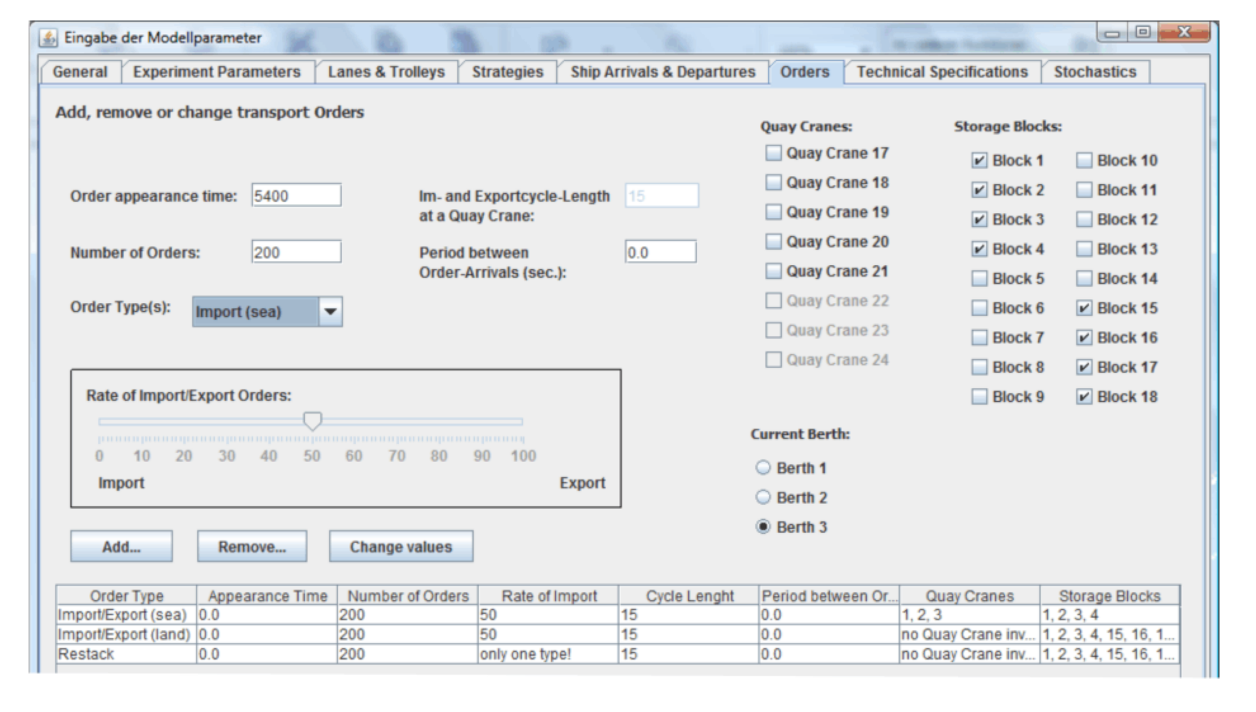


Abb. 6.4.: **Parametrisierung von Transportaufträgen**

6.2. statistischen Auswertungskomponenten des Modells

Um das Horizontalkrankonzept zu untersuchen, muss das Modell in der Lage sein, Kennzahlen zu liefern, die eine Vergleichbarkeit mit bestehenden Containerterminals ermöglichen. Desmo-J erzeugt für jeden Simulationslauf automatisch Dateien zur Auswertung, nämlich einen Trace und einen Report. Der Trace zeigt für `Event`s und `SimProcess`e die durch den Scheduler gesteuerten Ereignisse bezüglich anderer `SimProcess`e und `Entity` s. Das umfasst das Eintreten eines Ereignisses, die Aktivierung, Passivierung und das zeitkonsumierende Hold eines Prozesses und das Einfügen und Entfernen von Prozessen und Entitäten in Warteschlangen. Für alle Entitäten und Prozesse wird bei ihrer Erzeugung festgelegt, ob sie im Trace angezeigt werden sollen. Durch die dort aufgezeichneten Interaktionen von Prozessen und Ereignissen lässt sich das Modellverhalten im Kleinen untersuchen und seine Steuerung Schritt für Schritt nachvollziehen. Quantitative Aussagen sind über den Trace nicht zu erlangen. Er kann so parametrisiert werden, dass er nur einen Teil eines Simulationslaufs verfolgt, über den sich das grundsätzliche Modellverhalten nachvollziehen lässt.

Statistische Informationen werden in Desmo-J automatisch von den Warteschlangen eines Modells gesammelt. Zu jeder Warteschlange des Modells, also allen `Queue`s, `ProcessQueue`s, `WaitQueue`s etc. wird angegeben, ob sie im Report aufgeführt werden sollen. Warteschlangen liefern Informationen zur aktuellen, maximalen und durchschittlichen Anzahl ihrer Elemente, wie viele Elemente sie schon durchlaufen haben sowie die durchschnittlichen Wartezeit der Elemente. Sie bieten damit einige Auswertungsmöglichkeiten, mit denen sich Teile eines Modellverhaltens gut nachvollziehen lassen.

Neben der Auswertung von Warteschlangen, zur Abbildung von Warteverhalten, etwa auf eine Containerbrücke dienen, bietet es sich in Desmo-J an, sie auch dort einzusetzen, wo sie ausschließlich der Gewinnung statistischer Daten dienen. So kann zum Beispiel ein Transportauftrag, wenn er in das System eintritt, in eine Warteschlange eingefügt und bei Verlassen wieder entfernt werden, um den Durchschnittsaufenthalt aller Transportaufträge zu erfahren.

Neben dem Instrument Warteschlange bietet Desmo-J noch verschiedene andere statistische Auswertungs-Instrumente wie Zähler, Objekte für Summen- und Durchschnittsfunktionen und statistische Objekte für Zeitreihen-Untersuchungen etc.

Für die Untersuchung des Containerterminal-Modells werden über diese allgemeinen Auswertungs-Werkzeuge hinaus weitere, spezifische Instrumente benötigt, um etwa für einen bestimmten Schiffstyp den durchschnittlichen Containertransport pro Stunde und pro Containerbrücke auszuwerten. Zu diesem Zweck werden weitere Auswertungs-Instrumente erstellt, die sich zum Teil aus den in Desmo-J bestehenden zusammensetzen.

Differenzierung der Auswertungskennzahlen

Zur differenzierten Untersuchung des vorliegenden Modells sollen unterschiedliche Kenngrößen gewonnen werden. Sie sollen die Leistungsfähigkeit des Modells in verschiedenen Aggregations- und Differenzierungsgraden abbilden und dabei unterschiedliche Kennzahlen liefern:

- Durchschnittswerte für die Bearbeitungs- und die Wartedauer für jeden Auftragstyp
- Schaffung von Kenngrößen für ein Containerschiff
- Absolute Abweichungen der Ladezeiten gegenüber einem Schiff des Typs des Referenzmodells
- Relative Abweichungen von Ladezeiten gegenüber dem Referenzmodell
- Aggregation der Ladezeiten und Abweichungen gegenüber dem Referenzmodell bezüglich eines Schiffstyps
- Abweichungen der Schiffsladezeiten gegenüber einem Schiffsplan
- Differenzierung der Transport- und Wartezeiten an einem Schiff nach Containerbrücken
- Containerumschlag pro Stunde für jeden Schiffstyp
- Containerumschlag pro Stunde je Containerbrücke für jeden Schiffstyp

Um für alle Transportaufträge Informationen über ihre Warte- und Bearbeitungsdauer zu gewinnen, wird jeder Auftrag bei seiner Ankunft einem Evaluationsobjekt übergeben. Zusätzlich wird der Beginn seiner Abarbeitungsphase und das Ende des Containertransportes dem Evaluationsobjekt mitgeteilt, so dass eine statistische Auswertung vorgenommen werden kann.

Für landseitige Im- und Exportaufträge und für Restacking-Aufträge werden alle Kennzahlen über die gesamte Menge von Aufträgen gewonnen. Deswegen wird für ihre Auswertung modellweit ein einziges Auswertungsobjekt vom Typ `UnboundOrderEvaluator` erzeugt. Die Auswertung seeseitiger Transportaufträge hat als Erkenntnisgegenstand auch einzelne Schiffe und die Aggregation mehrerer Schiffe eines Typs, um darüber Aussagen treffen zu können. Deswegen werden für jedes einzelne Schiff Auswertungsobjekte vom Typ `VesselEvalutor` erzeugt.

Aus Sicht des Auftrags spielt es dabei keine Rolle, von welcher Art Auswertungsobjekt ihre Evaluationen vorgenommen werden. Für sie ist nur der Obertyp `Evaluator` (Abb. 6.5) bekannt. Damit Auswertungen über die einzelnen `Evaluator`-Objekte hinaus aggregiert erstellt werden können, werden alle Auswertungsobjekte zentral in einem `ExperimentEvaluator` gehalten.

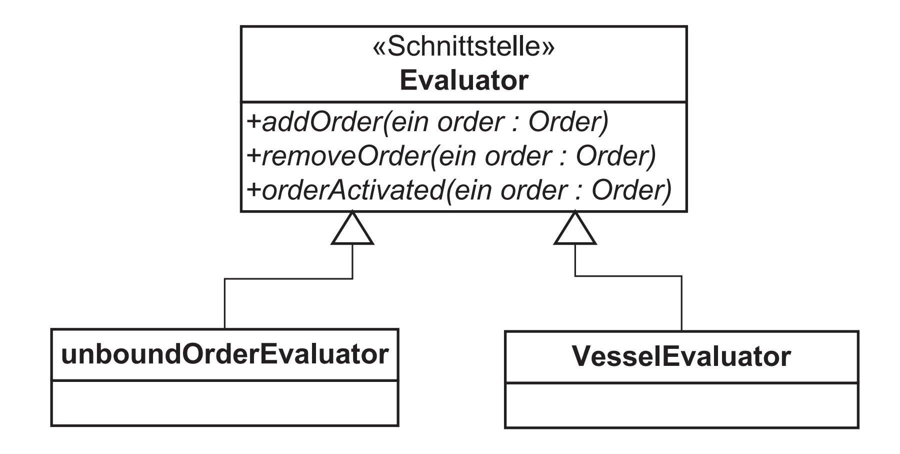


Abb. 6.5.: **Verwendung des Typs Evaluator**

6.2.1. Auswertung nicht schiffsgebundener Transportaufträge

Alle nicht schiffsgebundenen Transportaufträge werden durch das `UnboundOrderEvalua-tor`-Objekt ausgewertet. Dazu erhalten die landseitigen Im- und Exportaufträge und Restacking-Aufträge bei ihrer Erzeugung diesen `Evaluator` als Attribut.

Intern besitzt der Evaluator eine Reihe von Warteschlangen, in die die Aufträge zur Auswertung eingefügt und entfernt werden. Dabei gibt es für jeden der Auftragstypen eine Warteschlange zur Auswertung ihres Gesamtaufenthalts im System, in die der Auftrag bei seinem `OrderArrivalEvent` eingefügt und am Ende seiner Bearbeitung wieder entfernt wird. Zur Bestimmung der reinen Transportdauer gibt es zusätzlich eine Warteschlange für jeden Auftragstyp, in dem die Aufträge über die Zeit ihrer Bearbeitung gehalten werden. Dort werden sie zum Zeitpunkt ihres `SystemEntranceEvent`s eingefügt und zum Ende ihrer Bearbeitung wieder entfernt. In gleicher Weise werden Warteschlangen für Aufträge verspäteter LKW und eine Warteschlange für alle landseitigen Aufträge angelegt und verwendet.

Zur Auswertung werden die in den Warteschlangen gesammelten statistischen Werte von außen zugreifbar gemacht, so dass sie in die statistische Auswertung des Modells einfließen können. Die dadurch gelieferten Kennzahlen beschreiben für jeden der drei Auftragstypen und für die landseitigen Im- und Exporte in zusammengefasster Form die Gesamtanzahl der Aufträge, die Anzahl der verspäteten Aufträge, die der abgearbeiteten Aufträge und durchschnittliche Transport- und Gesamt-Aufenthaltsdauern.

6.2.2. Auswertung schiffsgebundener Transportaufträge

Bei der Auftragserzeugung durch den `InputToOrderConverter` wird für jedes in der Parametrisierung spezifizierte Containerschiff ein `VesselEvaluator` erzeugt. Alle Aufträge, die an dieses Schiff gebunden parametrisiert sind, erhalten die für dieses Schiff erzeugte Auswertungskomponente als Attribut.

Ein `VesselEvaluator` besitzt genau wie der Evaluator für die nicht schiffgebundenen Aufträge für alle Auftragstypen Warteschlangen. In diese werden unter den gleichen Bedingungen wie beim `unboundOrderEvaluator` Aufträge eingefügt und entfernt.

Neben den Warteschlangen zur statistischen Auswertung der Transportaufträge enthält der `VesselEvaluator` Informationen, die das Schiff betreffen. Sie werden bei der Erzeugung durch den `InputToOrderConverter` übergeben. Diese umfassen Daten, die einen rein beschreibenden Charakter für ein einzelnes Schiff besitzen, nämlich die fortlaufende Nummer des Containerschiffs im Modell und den Liegeplatz. Des weiteren dient der Schiffstyp dazu, dass alle `VesselEvaluator`en desselben Typs zusammengefasst angesprochen und ausgewertet werden können. Letztlich liegen noch Informationen vor, die direkten Einfluss auf die statistische Auswertung eines Schiffes haben. Das sind der Ankunftszeitpunkt und der erwartete Beladungs-Endzeitpunkt, der den Wert des Referenzmodells darstellt. Über diese beiden Werte wird eine erwartete Ladedauer ermittelt, über die sich relative und absolute Abweichungen gegenüber einem Vergleichsmodell ermitteln lassen. Daneben dient die Anzahl der Containerbrücken dazu, am Containerschiff Durchschnittswerte pro Containerbrücke, zum Beispiel für den Containerumschlag, zu generieren.

Zusätzlich wird innerhalb des `VesselEvaluator`s für jede an der Ladung beteiligte Containerbrücke ein Auswertungsobjekt erstellt, das wiederum Warteschlangen für Import- und Exportaufträge sowie beide Auftragstypen für jede Containerbrücke enthält und der Differenzierung der Transporte nach Containerbrücken dient.

Mit den statistischen Auswertungen durch die Warteschlangen und die Eingangswerte zum Containerschiff lassen sich nach einem Simulationslauf folgende statistische Kennzahlen für ein Schiff gewinnen:

- Durchschnittliche Wartezeit pro Import- und Export-Auftrag sowie für beide Auftragstypen zusammen
- Durchschnittliche Transportzeiten pro Import- und Export-Auftrag sowie für beide zusammen
- Gesamt-Containerumschlag pro Stunde
- Anzahl der Auftragstypen
- Ladezeit des Schiffes
- Absolute Abweichungen gegenüber dem Vergleichsmodell

- Relative Abweichungen gegenüber dem Vergleichsmodell
- Differenzierungen nach den einzelnen Containerbrücken
  - jeweilige Anzahl der Import-, Export- und Gesamtaufträge
  - Durchschnittsdauer und Wartezeit für jeden Auftragstyp
  - Gesamtladezeit an der Containerbrücke

Aus diesen Einzelkennzahlen lassen sich weitere Kennziffern gewinnen.

6.2.3. Ausgabe der Auswertung in Excel

Für jeden Simulationslauf wird die Auswertung der Ergebnisse in Form einer Excel-Datei ausgegeben. Diese enthält zusätzlich alle vorgenommenen Parametrisierungen, um das Simulationsexperiment umfassend zu beschreiben.

Die Modellparameter werden in sieben Arbeitsblättern nach Parametertypen unterteilt ausgegeben. Diese sind:

1. Experiment-Parameter
2. Ein- und Ausgabedateien
3. Dimensionierung der Low-Frame-Bridge
4. Nicht-schiffgebundene Auftragslisten
5. Die Schiffsliste und für jedes Schiff die daran gebundenen seeseitigen Im- und Exportaufträge
6. Technische Spezifikationen der Betriebsmittel
7. Eingaben zu Stochastik und Verspätungen

Anschließend werden die Simulationsergebnisse in gegliederter Form ausgegenben:

8. *Auswertungen nach Containerbrücken:* Es wird für jede Containerbrücke die Gesamt-Anzahl der abzuarbeitenden Transportaufträge und die Anzahl der nicht abgearbeiteten Aufträge am Ende des Simulationslaufs angegeben. Des weiteren werden die durchschnittliche Warte- und Bearbeitungszeit für einen Transportauftrag angegeben. Zur Evaluation dieser Werte wird auf die Warteschlangen der Containerbrücken zurückgegriffen.

9. *Auswertungen für jedes Containerschiff:* Die Ergebnisse jedes einzelnen `VesselEvaluator`s werden unter Nennung des Schiffs und seiner Attribute ausgegeben.

10. *Auswertungen nach Schiffstypen:* Für jeden Schiffstyp werden Durchschnitts- und absolute Werte bezüglich der dort verfügbaren Kennzahlen ausgegeben. Dazu werden die Kennzahlen über alle `VesselEvaluator`en eines Typs summiert und dann durch die Anzahl der Schiffe des Typs geteilt.

11. *Nicht schiffs-gebundene Aufträge:* Unter Nutzung des `UnboundOrderEvaluator`s werden die dort verfügbaren statistischen Werte zu landseitigen Im- und Exportaufträgen und Restacking-Aufträgen aufgeführt.

12. *Zusammenfassung:* Über alle vorliegenden `Evaluator`-Objekte werden absolute Werte und Durchschnittswerte bezüglich aller Transportaufträge ausgegeben. Des weiteren werden Auswertungen über alle seeseitigen Im- und Exportaufträge erstellt und Werte über die Ladezeiten aller Schiffe und ihren relativen und absoluten Abweichungen gegenüber dem Vergleichsmodell ausgegeben.

6.3. Animation der Simulationsergebnisse

Die Animationsdatei der Simulation baut auf ihrem Modell-Layout (Abschnitt 4.6) auf. Dort wird das Modell zunächst zu seinem Erzeugungszeitpunkt beschrieben. Während der Simulation treten weitere Animationsereignisse auf, um welche die Layout-Datei sukzessive erweitert wird.

Analog zum ersten Element „time“ wird dazu dem Layout-Dokument im Element „Viewport“ jeweils ein weiteres Zeit-Element mit dem Wert des Ereigniszeitpunkts zugefügt.

Dieses enthält wiederum Objekte, die durch ihre ID beschrieben werden und die im zuvor angegebenen Zeitpunkt ihren Zustand ändern. Die Zustandsänderungen werden durch die Kindknoten des Elements beschrieben. Die Animation der Simulationsergebnisse wird durch die Veränderung der Sichtbarkeit der verwendeten 3D-Komponenten, sowie der Rotation und der Bewegung zwischen zwei Punkten über einen Zeitraum vorgenommen (Abschnitt A.1). Dazu wird im Falle der Sichtbarkeitsänderung ein Element „visible“ mit einem Wahrheitswert angegebeben. Bei Rotation und Bewegung wird jeweils ein Element der Dauer und zwei Koordinaten-Tupel mit Ausgangs- und Zielpunkt bzw. Ausrichtung angehängt. Eine vollständige Beschreibung aller Möglichkeiten der Animation befindet sich in der Golem 3D Schittstellenspezifikation [ISL08].

6.3.1. Technische Realisierung der Animationsausgabe

Das Einlesen des Modell-Layouts wurde über jDom realisiert (Abschnitt 5.1). Dabei wird das gesamte Dokument im Speicher vorgehalten. Für große Dokumente ist dies aufgrund des hohen Ressourcenbedarfs nicht praktikabel. Schon die Animation des Be- und Entladens eines einzelnen großen Containerschiffs mit 3000 Transportaufträgen ohne parallel laufende Restacking- und Gate-Aktivitäten führt zu einer Dateigröße von über 30 MB. Deswegen wurde die Ausgabe über eine sequenzielle Verarbeitung realisiert, bei der das aktuelle Animationsereignis sofort in die Ausgabedatei geschrieben und nicht zwischengespeichert wird.

Verwendung gefunden hat dafür die „Simple Api for XML“, SAX. Sie ist ein Quasistandard für das Parsen von XML-Dateien ([ME07], S.41ff). Der SAX-Parser arbeitet ereignisorientiert. Das heißt er durchläuft ein XML-Dokument und erzeugt für alle unterschiedlichen Bestandteile des Dokuments, z.B. den Beginn oder das Ende eines Elementes, Textpassagen etc., verschiedene Ereignisse. In einer weiteren Komponente, dem „Content Handler“, wird dann gemäß den anwendungsspezifischen Anforderungen beschrieben, wie das Programm auf ein bestimmtes Ereignis reagiert.

Für die Erzeugung der Animationsdatei wurde in folgender dreischrittiger Weise vorgegangen:

1. Zur Vorbereitung der Ausgabe der Animationsereignisse durchläuft der Parser zunächst das Modell-Layout vollständig. Der „Content Handler“ schreibt den gesamten Inhalt in eine neue Datei bis zu dem Punkt, an dem die Animationsereignisse eingefügt werden sollen; also unterhalb des „time“-Elements, in dem die Modellkomponenten initialisiert werden.

2. Im zweiten Schritt werden während des Simulationslaufs die XML-Repräsentationen der Animationsereignisse in das Dokument geschrieben.

3. Nach Beendigung der Simulation wird das Layout-Dokument noch einmal geparst. Der ContentHandler kopiert diesmal den abschließenden Teil des Layout-Dokumentes in die neue Animationsdatei und schließt diese ab.

6.3.2. Darstellung der Animationsausgabe in Golem 3D

Die während der Simulation erzeugte Animationsdatei lässt sich nach Ende des Simulationslaufs im 3D Realtime Animation Player Golem 3D [ISLa] abspielen (Abb. 6.6). Durch die grafische Darstellung werden manche Modellzusammenhänge intuitiv fassbar. Bestimmte, insbesondere räumliche, Inkonsistenzen des Modells sind dabei zum Teil sehr viel leichter auszumachen, als bei einer höheren, zum Beispiel auf reine Koordinaten abgebildeten, Abstraktionsebene des Modellverhaltens. Während die Überschneidung zweier Containerbrücken bei einer nicht-grafischen Systemrepräsentation unter Umständen nicht auffällt, wird sie im Rahmen einer Animation leicht sichtbar.

Zur Untersuchung bestimmter Modellbereiche oder Interaktionen zwischen Komponenten, die auf das Modellverhalten besonderen Einfluss haben, gibt es in Golem 3D die Möglichkeit, die Kameraperspektive der Animation frei über das gesamte Modell zu wählen und während der Animation zu verändern und zu zoomen. Die Bewegung einzelner Objekte lässt sich mit der Einstellungsmöglichkeit nachvollziehen, die Kamera entweder dem Objekt folgen zu lassen oder sie direkt daran zu binden. Die genaue Beobachtung bestimmter Zeiträume ist mit einer Beschleunigungs- und einer Zeitlupenfunktion sowie dem Springen an bestimmte Zeitpunkte der Animation möglich. Zur Präsentation von Animationsergebnissen erlaubt es Golem 3D darüber hinaus, Kamerafahrten explizit zu defineren und Szenen zu einem Film zusammenzuschneiden.

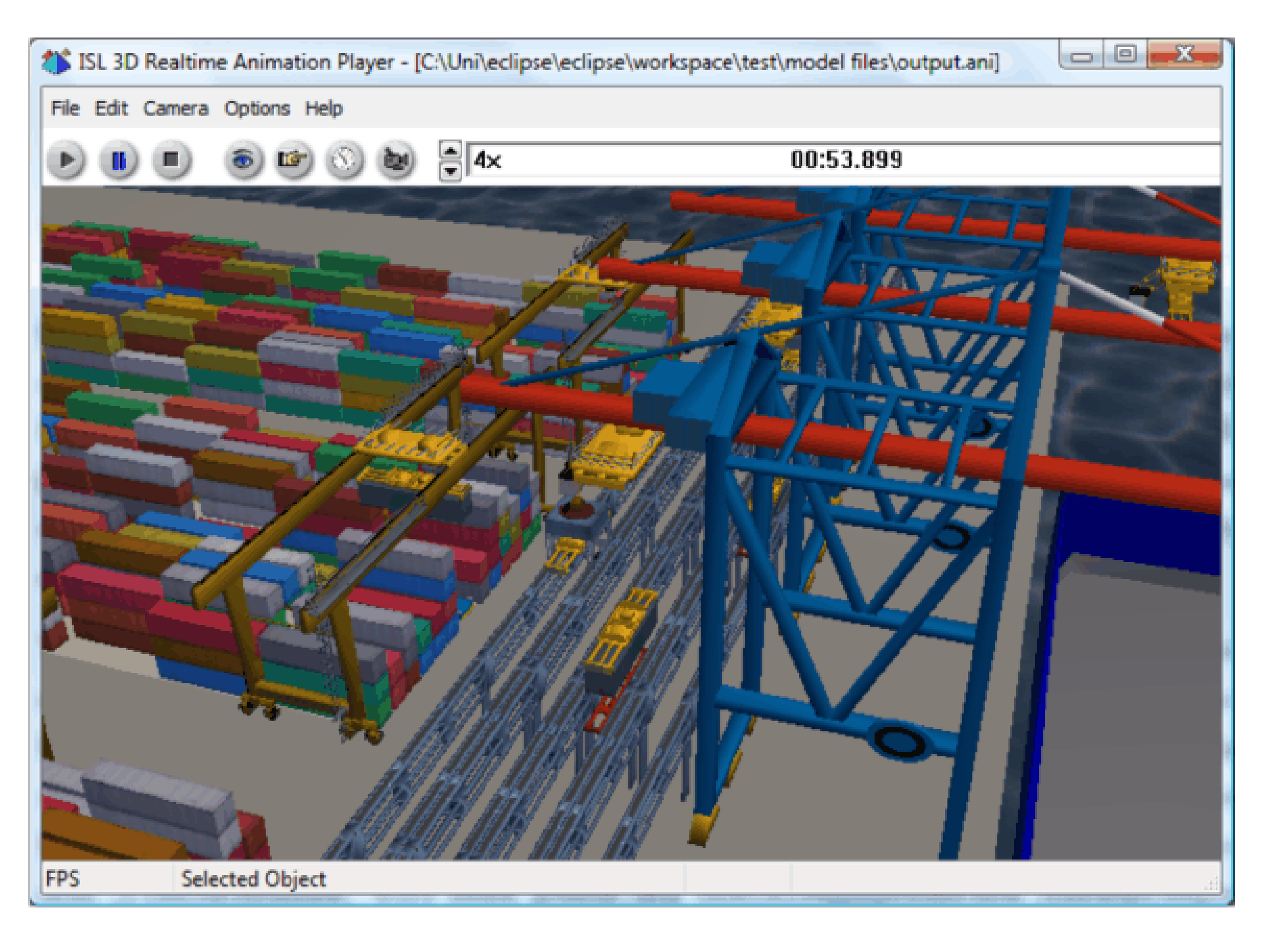


Abb. 6.6.: **Animation in Golem 3D**

7. Auswertung des Horizontalkrankonzepts

Die Analyse des Horizontalkrankonzeptes wurde auf der Grundlage eines Vergleichs des im Rahmen dieser Arbeit realisierten Modells mit einem am ISL erstellten Modell eines Containerterminals mit Van-Carrier-Architektur durchgeführt. Bei der Modellerstellung wurde daher darauf geachtet, eine ähnliche Dimensionierung zu erreichen. Das vorliegende Referenzmodell enthält Spezifikationen für Schiffsankünfte. Die dort unterschiedenen Typen von Containerschiffen werden bei der Erstellung dieses Modells abgebildet, so dass sie als Teil eines Szenarios definiert werden konnten. Das Ausgangsmodell gibt für jeden dieser Schiffstypen eine feste Anzahl Im- und Exportaufträge, die während der Liegezeit durchgeführt werden und die dafür benötigte Ladezeit, an. Darüber hinaus wird die durchschnittliche Schiffsankunftsrate pro Woche für jeden Typ bestimmt. Es wurde weiterhin eine nicht-produktive Liegezeit von zwei Stunden für jedes Containerschiff angenommen. Daneben sind für jeden Wochentag eine Anzahl Restacking-Aufträge angegeben. Die landseitigen Im- und Exporte sind für jeden Wochentag nach Acht-Stunden-Schichten für LKW- und Eisenbahn-Gate getrennt spezifiziert.

Um eine aussagekräftige Bewertung des Horizontalkrankonzepts vornehmen zu können, wird zunächst ein Szenario mit einem vollen Wochenplan modelliert. Dazu werden die Restacking-Aufträge einer Woche gleichmäßig über die gesamte Woche verteilt. Die Im- und Exportaufträge an den Eisenbahn- und LKW-Gates des Referenzmodells werden bei der Parametrisierung des Szenarios über die Werktage und LKW-Gates verteilt. Die durchschnittliche Anzahl von Containerschiffen wird aus dem Referenzmodell für jeden Typ gerundet und möglichst gleichmäßig über die Woche verteilt in Schiffsankünfte des Szenarios abgebildet. Die Anzahl der Im- und Exportaufträge wird dem Referenzmo-

dell entsprechend gewählt und die Anzahl der Containerbrücken mit einem realistischen Standardwert für jedes Schiff belegt.

Da Desmo-J (noch) keine Konvertierung von Zeitformaten anbietet, wird zur Darstellung aller Zeitpunkte die Grundeinheit Sekunde verwendet. Eine eigenständige Konvertierung von Zeitformaten wurde nicht realisiert, da jene Kennzahlen, die einen bewertenden Vergleich erlauben in Form von Prozentangaben ausgegeben werden.

7.1. Simulation eines Wochenplans

Der konkrete Wochenplan, der zur Gewinnung eines bewertenden Vergleichs simuliert werden soll, enthält die folgende Parametrisierung:

Für die Systemeingangs-Strategie wurde „Strict Order at the Quay Cranes" gewählt. Diese Steuerungsregel erschien am sinnvollsten, weil sie sowohl die Reihenfolge-Beziehungen der Schiffsstaupläne berücksichtigt als auch eine Abarbeitung der übrigen Aufträge unabhängig von ihrer Ankunftsreihenfolge erlaubt.

Die Dimensionierung der Low-Frame-Bridge wurde mit allen vier Spuren mit je vier Flat- und sieben Lifting-Trolleys gewählt.

In das Szenario wurden die Schiffsankünfte des Refenzmodells modelliert, diese sind:

- 10 Feederschiffe des Typs F480 mit je 240 Im- und Export-Aufträgen, 3 Containerbrücken und einer Referenzladezeit von 8,5 Stunden.
- 7 Feederschiffe des Typs F670 mit je 335 Im- und Export-Aufträgen, 3 Containerbrücken und einer Referenzladezeit von 11,9 Stunden.
- 8 Containerschiffe des Typs M1600 mit je 800 Im- und Export-Aufträgen, 5 Containerbrücken und einer Referenzladezeit von 10,1 Stunden.
- 3 Containerschiffe des Typs J2400 mit je 1200 Im- und Exportaufträgen, 6 Containerbrücken und einer Referenzladezeit von 12,4 Stunden.
- 1 Containerschiff des Typs J3000 mit je 1500 Im- und Exportaufträgen, 6 Containerbrücken und einer Referenzladezeit von 15,5 Stunden

Zur Abbildung der Restacking-Aufträge des Referenzmodells werden 980 Aufträge dieses Typs angelegt, die verteilt über die Woche alle 10 Minuten am System ankommen.

Sowohl die Aufträge der LKW- und als auch die des Eisenbahn-Gates des Referenzmodells werden auf Transporte am LKW-Gate abgebildet. Dabei wurden im Szenario die 8700 LKW-Transporte durch eine landseitige Auftragsankunft alle 50 s und 4350 Eisenbahn-Transporte durch eine landseitige Auftragsankunft alle 100 s abgebildet.

Die Zuordnung der Stacks im Yard zu den land- und seeseitigen Im- und Exportaufträgen wird zufällig über alle Containerblöcke vorgenommen.

Auswertung eines Wochenplan-Szenarios

Die Auswertung des Simulationexperiments ergibt über den gesamten Wochenplan eine durchschnittliche Abweichung der Ladedauer von gut 1000%[1]. Die Kennziffer „Ladedauer“ wird so definiert, dass sie den Zeitraum zwischen geplantem Beginn der Be- und Entladung eines Containerschiffs und dem Zeitpunkt beschreibt, an dem der letzte Transportauftrag des Schiffes abgearbeitet ist, beschreibt. Wenn die Ladung eines Containerschiffs länger dauert und sich damit der Beginn der Bearbeitung des nächsten verschiebt, erhöht sich damit in der vorliegenden Modellierung auch dessen Ladezeit. Die Abweichung der Ladezeit ergibt sich aus der Differenz zwischen Ladezeit und Referenzladedauer des Schiffstyps. Im Arbeitsblatt „Vessel Evaluation“ der Auswertungsdatei lässt sich dazu erkennen, dass die ersten Containerschiffe eine verlängerte Ladedauer von etwa 100% gegenüber dem Referenzmodell haben, während diese bei den letzten schon um die 3000% beträgt. Neben der Differenz des Endes der erwarteten Ladezeit und dem wirklichen Ende ist auch die um Verspätungen bereinigte Nettoladezeit der verschiedenen Schiffstypen um ein Mehrfaches höher als beim Vergleichsmodell.

Aufgrund der enorm längeren Ladezeit gegenüber dem Referenzmodell ist das Horizontalkrankonzept in der vorliegenden Modellierung nicht geeignet, eine Alternative gegenüber bestehenden Containerterminal-Architekturen zu bieten. Die Experimentdatei ist der Arbeit unter den Namen „vollerWochenplan.xls“ beigefügt.

[1]Die Excel-Dateien zu den einzelnen Experimenten liegen der Diplomarbeit in elektronischer Form bei.

7.2. Annahme von Tandem-Lift

Im Präsentationsfilm der Firma ZPMC [ZPMb] sind alle Betriebsmittel des Containerterminals auf den parallelen Transport zweier 40 ft Container gleichzeitig ausgelegt. Das bedeutet, dass die Conteinerbrücken mit zwei Spreadern ausgerüstet sind, die beide über ein eigenes Hubwerk verfügen und zwei Container gleichzeitig auf ein oder von einem Containerschiff laden können. Die Flat-Trolleys und die Terminal-Chassis haben beide die doppelte Breite und die Lifting-Trolleys können zwei Container in einem Arbeitsschritt synchron zwischen Flat-Trolleys und Terminal-Chassis umladen. Ebenso verfügen die Portalkrane über zwei unabhängige Katzen, die jeweils ein eigenes Hubwerk mit Spreader besitzen.

Unter der Prämisse, dass die Stellplatzplanung in der Lage ist, einen entsprechenden Vorstau zu leisten und der Abstraktion der dadurch entstehenden Mehrarbeit sowie der Vernachlässigung der weiteren Fahrwege durch breitere Querspuren, kann angenommen werden, dass die passenden Betriebsmittel bei jedem Containertransport zwei 40 ft Boxen bewegen können. Damit wären, insbesondere auf der Low-Frame-Bridge und an den Containerbrücken, nur die Hälfte der Transportaufträge nötig.

Zur Untersuchung des Beispielterminals unter diesen Annahmen wird ein weiterer Wochenplan mit denselben Parametrisierungen des Szenarios simuliert, nur dass lediglich die Hälfte der seeseitigen Transportaufträge angenommen wird.

Auswertung eines Tandem-Lift-Szenarios

Unter den Annahmen zum Betrieb des Terminals mit Tandem-Lift beträgt die durchschnittliche Abweichung der Ladezeit gegenüber dem Referenzmodell 280%. Die Experimentdatei liegt der Arbeit unter dem Namen „TandemLift.xls“ bei.

Also stellt das Horizontalkrankonzept auch unter den dort getroffenen Annahmen keine Alternative zu bestehenden Containerterminal-Architekturen dar. Das könnte verschiedene Gründe haben, die im Folgenden mit weiteren Simulationsexperimenten untersucht werden:

- Die gleichzeitige Durchführung landseitiger und seeseitiger Im- und Exportaufträge im selben Stack könnte dort die Ressource Portalkran zu stark binden. Dies könnte entweder durch eine zu wenig differenzierte Abbildung der Steuerung, die zu umfassende Reservierungen an den Portalkranen vornimmt oder an einer zu gering gewählten Dimensionierung der Betriebsmittel im Stack liegen.

- Die abgebildete Steuerung verteilt die Containerstellplätze für die Transportaufträge über alle Stacks des Terminals. Dadurch entstehen weite Wege die auf der Low-Frame-Bridge zurückzulegen sind und gegenseitige Blockierungen zwischen den Flat-Trolleys bei der parallelen Bearbeitung zweier Containerschiffe.

- Die technischen Spezifikationen der Betriebsmittel des Horizontalkrankonzepts, Flat- und Lifting-Trolley und Terminal-Chassis lagen nicht vor und wurden daher geschätzt. Andere als die gewählten Werte könnten die Low-Frame-Bridge als möglichen Flaschenhals abmildern oder entschärfen.

- Das Horizontalkrankonzept als Architektur für Containerterminals kann insgesamt nicht geeignet sein.

7.2.1. Untersuchung eines Tandem-Lift-Szenarios ohne landseitige Transporte

Um den Einfluss der landseitigen Im- und Exportaufträge auf die Leistungsfähigkeit des Containerterminals bezüglich des Umschlages an den Containerbrücken zu untersuchen, wird der Wochenplan des Tandem-Lift-Szenarios ohne landseitige Transportaufträge untersucht. Die Annahme, dass ein Stack, der an der Be- und Entladung von Containerschiffen beteiligt ist, von landseitigen Transporten ausgenommen ist, erscheint nicht realistisch, jedoch könnte durch die Stellplatzplanung mit passenden Restacking-Aufträgen das Maß an gleichzeitigen land- und seeseitigen Transportaufträgen an einem Stack vermindert werden.

Das zu der Untersuchung dieser Annahme erstellte Szenario entspricht genau dem Tandem-Lift-Szenario, nur dass auf die Abbildung der LKW Im- und Exportaufträge verzichtet wurde.

Auswertung des „Szenarios ohne LKW-Transporte“

Die Veränderung der durchschnittlichen Ladezeiten ist durch den Verzicht der Abbildung von landseitigen Transporten nur geringfügig. Die durchschnittliche Verkürzung der Ladezeiten über alle Schiffe des Wochenplans beträgt knapp 3%. Dieses ist insbesondere durch eine Priorisierung in der Modellsteuerung begründet. Die landseitigen Aufträge eines Szenarios erreichen in der Regel nach den seeseitigen das System, da die schiffsgebundenen Aufträge mit der Ankunft des Schiffes komplett vorliegen und die LKW-Ankünfte über den abgebildeten Zeitraum verteilt sind. Wenn Ressourcen durch einen land- und einen seeseitigen Auftrag angefragt werden, so erhält der ältere Auftrag seine Freigabe, was zumeist der seeseitige Transportauftrag ist. Verzögerungen der seeseitigen Im- und Exportaufträge treten damit nur auf, wenn während eines LKW Im- oder Exports fehlende Ressourcen für die Bearbeitung des seeseitigen Auftrags freiwerden und umfassen die verbleibende Bearbeitungsdauer des LKW-Transportauftrags. Die gewählte Modellierung scheint geeignet in realistischer und effizienter Weise die Gate-Operationen des Terminals abzubilden. Die Experimentdatei liegt unter dem Namen „TandemLiftKeineLandseite.xls“ der Arbeit bei.

7.2.2. Untersuchung eines Szenarios mit fest zugewiesenen Lagerblöcken

In den bisherigen Szenarien wurde davon ausgegangen, dass die Im- und Exportaufträge eines Containerschiffes zwischen einer Containerbrücke und einem beliebigen Stack im Yard durchgeführt werden. Die Annahme eines stärkeren Vorstaus, der alle auf ein Schiff zu transportierenden Container in bestimmten Blöcken des Yard vorhält, erscheint jedoch durchaus realistisch. Deswegen wird ein Szenario untersucht, in dem alle seeseitigen Im- und Exportaufträge ausschließlich sechs dem Liegeplatz am nächsten gelegenen Stacks umfassen. Ziel dieses Experimentes ist es, herauszufinden, in wieweit die kürzeren auf der Low-Frame-Bridge zurückzulegenden Wege und die Reduzierung der gegenseitigen Blockierung von Flat-Trolleys, die parallel zwei Schiffe be- und entladen Einfluss auf die Effizienz des Modells haben. Dazu wird das Tandem-Lift-Szenario derart modifiziert, dass die Containerslots der Transportaufträge eines Schiffes nur über die acht dem Liegeplatz am nächsten gelegenen Stacks verteilt generiert werden.

Auswertung des Szenarios mit fest zugewiesenen Lagerblöcken

Die Festlegung der seeseitigen Transportaufträge auf einen bestimmten, nahe gelegenen Bereich führt zu einer erheblichen Verkürzung der Schiffsladezeiten. Die reinen Ladezeiten der Feederschiffe sind hier gegenüber dem Referenzmodell um durchschnittlich 27% beim Feeder F-480 und 31% beim Feeder F-670 verkürzt. Die größeren Containerschiffe haben jedoch eine längere reine Ladezeit: Beim Typ M-1600 um 40%, beim Typ J-2400 um 59% und beim Typ J-3000 um 41%.

Das schnellere Be- und Entladen der Feederschiffe erklärt sich dadurch, dass sie relativ zu der Anzahl ihrer Container lange Referenzladezeiten besitzen, von vielen Containerbrücken bedient werden und die Dimensionierung der Low-Frame-Bridge und ihrer Betriebsmittel für alle Schiffe gleich ist. Für die absoluten Ladezeiten des Wochenplans gegenüber dem Referenzmodell, also unter Nichtberücksichtigung verspäteter Anfangszeiten durch längere Ladezeiten der vorherigen Containerschiffe eines Liegeplatzes ergibt sich eine längere Ladedauer von 80%. Die Auswertungsdatei zu diesem Experiment liegt unter dem Namen „TandemLiftFesteStackzuordnung.xls“ vor.

7.2.3. Untersuchung eines Szenarios mit veränderten Flat- und Lifting-Trolley-Geschwindigkeiten

Für die Modellierung des Containerterminal-Modells liegen keine technischen Spezifikationen des Flat- und Lifting-Trolleys sowie des Terminal-Chassis vor. Die Werte für die Fahrgeschwindigkeiten dieser Komponenten sowie der Hub- und Drehgeschwindigkeit des Lifting-Trolleyspreaders werden geschätzt. Diese Werte liegen als Default-Werte zur Szenario-Parametrisierung vor und werden bisher in allen diskutierten Szenarien verwendet. Sie werden mit 18 km/h als Fahrgeschwindigkeit für Flat-Trolleys und Terminal-Chassis angenommen und 60 m/min für die Lifting-Trolleys. Für die Hub-Geschwindigkeit der Lifting-Trolleys wurden 50 m/min und für das Drehen des Spreaders 5 s verwendet. Die getroffenen Annahmen sind insgesamt recht vorsichtige Schätzungen. In einem weiteren Szenario werden andere Annahmen bezüglich der Geschwindigkeiten der Betriebsmittel des Horizontalkrankonzepts untersucht.

Dazu wird mangels Realdaten für die Geschwindigkeiten des Lifting-Trolleys die Laufkatzengeschwindigkeit der Containerbrücke, also 220 m/min angenommen. Es handelt sich in beiden Fällen um eine Schienenkonstruktion, die auf Stahlstreben geführt ist und in der Lage ist, schwere Lasten zu heben und zu befördern. Die Fahrgeschwindigkeit der ebenfalls schienengeführten Terminal-Chassis wird mit der Geschwindigkeit der Laufkatze der Conatainerbrücke, die ebenso schienengeführt ist, parametrisiert. Für die Geschwindigkeiten der Flat-Trolleys und der Terminal-Chassis werden die der Geradeausfahr-Geschwindigkeit der AGV der Firma Gottwald [Gotb] mit 6 m/s angenommen. Da die AGV rad- und nicht schienengeführt sind, könnte wegen des geringeren Reibungswiderstandes etwas höhere Geschwindigkeiten angenommen werden, allerdings wird von Beschleunigungs- und Bremsgeschwindigkeiten abgesehen, so dass die getroffene Annahme begründbar erscheint.

Die übrige Parametrisierung entspricht der des Szenarios mit Tandem-Lift bei fester Zuordnung der Stacks zu einem Schiff.

Auswertung des Szenarios mit veränderten Geschwindigkeiten

Die Erhöhung der Geschwindigkeiten der Modellkomponenten der Low-Frame-Bridge führt zu einer Verkürzung der reinen Schiffsladezeiten von etwa 5%. Mit dieser Verbesserung dauert das Be- und Entladen der nicht-Feeder Conainerschiffe etwa 40% länger als im Referenzmodell. Unter den gewählten Parametrisierungen stellt das Horizontalkrankonzept keine Alternative zu bestehenden Terminal-Architekturen dar. Die Experimentdatei zu diesem Szenario liegt der vorliegenden Arbeit unter dem Namen „TandemLiftFesteStackzuordnungHöhereGeschwindiekeit.xls“ bei.

7.2.4. Untersuchung der Be- und Entladung einzelner Containerschiffe

Um die Einflüsse des gleichzeitigen Be- und Entladens von Containerschiffen innerhalb des Wochenplans auf die Ladedauer der einzelnen Schiffe festzustellen, wird jeweils ein Schiff jedes Typs einzeln untersucht. Dabei bildet das Szenario zur Abbildung von Tandem-Lift und fester Zuordnung eines Yard-Bereichs mit den Default-Werten der

technischen Spezifikation den Rahmen. Das Auftreten der landseitigen Transportaufträge und der Restacking-Aufträge wird beibehalten. Die jeweilige Experimentdatei liegt der vorliegenden Arbeit unter dem Namen ihres Schiffstyps bei.

Zu Beginn der Bearbeitung eines Schiffes verteilen sich die Lifting-Trolleys über die genutzten Bereich der Low-Frame-Bridge. Diese Aufwärmphase bringt eine leichte Verlängerung der ersten Transportaufträge mit sich.

Untersuchung einzelner Schiffsankünfte je eines Schiffstyps

Das Experiment am Szenario eines Feeder-Schiffes des Typs F-480 ergab eine um 27% verkürzte Ladezeit gegenüber dem Sollwert. Ebenso ergab die Auswertung eines einzelnen F-670 mit -29%, eines M-1600 mit + 49%, des J-2400 mit + 61% und des J-3000 mit +61% Abweichung gegenüber der Referenz-Ladedauer keine großen Änderungen im Vergleich zur Bearbeitung mehrerer Schiffe gleichzeitig in einem Wochenplan. Die Erklärung dafür, dass innerhalb eines Wochenplans die gleichzeitige Bearbeitung mehrerer Containerschiffe und ihrer Im- und Exportaufträge keine Effizienzeinbuße bedeutet, ist, dass die Wege zwischen den Schiffen und den jeweiligen Stacks sich nicht überschneiden und die Anzahl der Flat- und Lifting-Trolleys ausreichend groß ist, so dass sie keinen Engpass darstellt.

8. Fazit

Das Horizontalkrankonzept konnte mit Desmo-J und Golem 3D in einem Beispielterminal modelliert und die einzelnen Simulationsläufe animiert und ausgewertet werden. Dazu wurde zunächst ein Containerterminal unter Verwendung des Horizontalkonzepts entworfen. Für dieses wurden Dimensionen gewählt, die einem echten Containerterminal möglichst nahe kommen, um einen realistischen Rahmen für die Untersuchungen zu bilden. Es wurden Abstraktionen von nicht modellrelevanten Systembereichen vorgenommen und verschiedene Annahmen bezüglich der Steuerung getroffen. Die grafischen Modellkomponenten wurden, soweit nicht verfügbar in einem 3D Design-Programm erstellt. Im Modell-Layout-Editor Verdigo wurde das Modell mit allen Komponenten entsprechend ihrer Größe und Lage im Modell definiert. Es wurde ein Simulationsmodell in Desmo-J erstellt, das auf die im Layout spezifizierten Komponenten und ihre Anordnung zugreift und die Steuerung des Modells spezifiziert. Für das Modell wurde eine grafische Benutzerschnittstelle modelliert, in der die Parametrisierung von Szenarien vorgenommen wird. Diese Szenarien spezifizieren Experimente, die in Desmo-J durchgeführt werden können. Nach Durchführung eines Experimentes liegt das spezifische Modellverhalten, das im Verlauf desselben auftrat, in Form einer Animationsdatei vor und kann in Golem 3D betrachtet und untersucht werden. Die Simulationsergebnisse des Experimentes liegen in Form einer Excel-Datei vor. Sie bieten umfangreiche statistische Kennzahlen zum Simulationslauf unter der angegebenen Szenario-Parametrisierung.

Im gewählten Aufbau und unter den gewählten Annahmen bezüglich Steuerung und technischer Parameter war es nicht möglich, die durch das Vergleichsterminal gegebenen Ladezeiten für größere Containerschiffe zu erreichen. Bei ihnen ist im Rahmen der modellierten Implikationen ein zeitlicher Mehraufwand von etwa 50% für das Be- und Entladen nötig.

Bei der Modellierung der Steuerung wurden verschiedene Vereinfachungen gegenüber einer realen Containerterminal-Steuerung verwendet, die Einfluss auf ihre Effizienz haben könnten. Bei einer weiteren Untersuchung des Horizontalkrankonzeps im Rahmen einer größeren Studie könnte eine Ausdifferenzierung dieser Abstraktionen die gewonnenen Aussagen weiter verfeinern bzw. verändern.

Die Stau- und die Stellplatzplanung wurden nicht explizit modelliert. Stattdessen muss der Benutzer des Simulationsmodells angeben, welche Bereiche des Yards an einer Gruppe von Transportaufträgen beteiligt sind. Bei Container-Transporten aus einem Lagerblock auf ein Schiff wird pauschal ein Vorstau in der dem Schiff zugewandten Seite des Blocks angenommen. Wird ein Containerschiff entladen, werden alle Container zufällig über den Block verteilt. Eine konkrete Stellplatzplanung ist notwendig, um die getroffenen Annahmen zu bestätigen oder ein abweichendes Verhalten zu zeigen. Weiterhin könnte die Güte des Vorstaus in Abhängigkeit der Auslastung bestimmter Ressourcen variieren, wenn untätige Betriebsmittel sukzessive über Restacking diesen verbessern. Eine Ad-hoc-Stellplatzzuweisung könnte kurzfristig freie Ressourcen antizipieren und so Wartezeiten vermindern.

In der vorliegenden Modellierung werden Containertransporte erst begonnen, wenn alle beteiligten Betriebsmittel, mit Ausnahme des Lifting-Trolleys, verfügbar sind. Abgesehen von den Containerbrücken unternehmen die beteiligten Betriebsmittel keine Teilschritte eines Transportes, bevor alle Betriebsmittel verfügbar sind. Dieses liegt darin begründet, dass die Steuerung erst kurzfristig bestimmt, welcher Transportauftrag der nächste abzuarbeitende ist. Eine differenziertere Steuerung, die prospektiv mehrere zukünftige Container-Transporte plant und dabei das frei werden der einzelnen Betriebsmittel antizipiert, könnte die Transportzeiten verringern.

Die Zuordnung, welche Containertransporte zu einem Zeitpunkt durchgeführt werden können wird über unterschiedliche, vereinfachende Regeln spezifiziert (Abschnitt 4.4). Eine explizite Stellplatz-Planung oder eine weitere Verfeinerung dieser Regeln könnte dafür sorgen, bestimmte Reihenfolge-Abhängigkeiten bzw. die Möglichkeit verschiedene Transporte in beliebiger Reihenfolge durchzuführen, weiter zu differenzieren und damit einen höheren Grad an Realismus zu erreichen.

Eine technische Spezifikation der beteiligten Containerterminal-Komponenten liegt nicht

vor. Sollten die technischen Eigenschaften stark von den hier getroffenen Annahmen abweichen, sollten weitere Untersuchungen durchgeführt werden.

Die Kosten für die im Horizontalkrankonzept verwendeten Komponenten sind nicht bekannt. Es ist jedoch zu vermuten, dass die große Anzahl Einzel-Krane und die feste Installation einer frei schwebenden mehrspurigen Stahlbrücke die auf den zügigen Transport von Containern und schwerem Kran-Gerät ausgelegt ist, hohe Investitionen erfordert, die oberhalb derer anderer stark automatisierter Terminals liegen. Ein weiteres Problem des Konzeptes ist vermutlich die schlechtere Wartbarkeit der Geräte gegenüber mobilen Systemen wie Van Carriern oder AGV. Ein großer Vorteil der in dieser Arbeit diskutierten Architektur, scheint die, durch die kompakte Anordnung der Systemkomponenten erreichte, sehr gute Raumausnutzung zu sein.

Ob das Horizontalkrankonzept letztlich in einem Containerterminal produktiv angewandt werden kann, müssen weitere Untersuchungen zeigen. Unter den im Rahmen dieser Arbeit getroffenen Annahmen scheint es keine sinnvolle Alternative zu bestehenden Containerterminal-Architekturen zu sein.

A. Anhang

A.1. Bewegung, Rotation und Sichtbarkeit in Golem 3D

```xml
<time value="268.97299999999996">
  <object id="CAR1-CT40" default.autorotate="0">
    <visible value="true"/>
  </object>
</time>
<time value="268.97299999999996">
  <object id="HORCRANE1-3-SP" default.autorotate="0">
    <move time="1.5">
      <org>0, 0, -7.25</org>
      <org>0, 0, -6.0</org>
    </move>
  </object>
</time>
<time value="270.47299999999996">
  <object id="HORCRANE1-3-SP" default.autorotate="0">
    <rotate time="5.0">
      <angles x="current" y="90.0" z="current"/>
      <angles x="0" y="0.0" z="0"/>
    </rotate>
  </object>
</time>
```

A.2. Beispiel einer vollständigen Szenariodatei

```xml
<?xml version="1.0" encoding="UTF-8"?>
<parameters>
  <experiment>
    <orderStrategy>3</orderStrategy>
    <experimentPeriod>
      <start>0.0</start>
      <finish>100000.0</finish>
```

```
8       </experimentPeriod>
9       <tracePeriod>
10        <start>0.0</start>
11        <finish>1000.0</finish>
12      </tracePeriod>
13      <debugPeriod>
14        <start>0.0</start>
15        <finish>1000.0</finish>
16      </debugPeriod>
17    </experiment>
18    <paths>
19      <scenarioFile>defaultParameters.xml</scenarioFile>
20      <modelInput>model files/modell-objects.ani</modelInput>
21      <modelOutputAni>model files/output.ani</modelOutputAni>
22      <generateAnimation>true</generateAnimation>
23      <extraAni>model files/extra.ani</extraAni>
24      <excelOutput>model files/output.xls</excelOutput>
25    </paths>
26    <model>
27      <lane1>
28        <visible>true</visible>
29        <numberFlatTrolleys>3</numberFlatTrolleys>
30        <numberLiftingTrolleys>7</numberLiftingTrolleys>
31      </lane1>
32      <lane2>
33        <visible>true</visible>
34        <numberFlatTrolleys>3</numberFlatTrolleys>
35        <numberLiftingTrolleys>7</numberLiftingTrolleys>
36      </lane2>
37      <lane3>
38        <visible>true</visible>
39        <numberFlatTrolleys>3</numberFlatTrolleys>
40        <numberLiftingTrolleys>7</numberLiftingTrolleys>
41      </lane3>
42      <lane4>
43        <visible>true</visible>
44        <numberFlatTrolleys>3</numberFlatTrolleys>
45        <numberLiftingTrolleys>6</numberLiftingTrolleys>
46      </lane4>
47    </model>
48    <nonSeasideOrders>
49      <orderList orderType="Import/Export␣(land)" appearanceTime="0.0" numberOfOrders="
            200" rateOfImport="50" cycleLenght="15" periodBetweenOrders="0.0"
            selectedQuayCranes="no␣Quay␣Crane␣involved!" selectedStorageBlocks="1,␣2,␣3,␣4,
            ␣15,␣16,␣17,␣18␣" />
50      <orderList orderType="Restack" appearanceTime="0.0" numberOfOrders="200"
            rateOfImport="only␣one␣type!" cycleLenght="15" periodBetweenOrders="0.0"
            selectedQuayCranes="no␣Quay␣Crane␣involved!" selectedStorageBlocks="1,␣2,␣3,␣4,
            ␣15,␣16,␣17,␣18␣" />
51    </nonSeasideOrders>
52    <vesselList>
```

```
53      <vessel berth="1" appearanceTime="0.0" estimatedEndOfLoading="30600.0"
            departureTime="32600.0" vesselType="F␣−␣480" numberOfQuayCranes="3">
54        <orderList orderType="Import/Export␣(sea)" appearanceTime="0.0" numberOfOrders="
              200" rateOfImport="50" cycleLenght="15" periodBetweenOrders="0.0"
              selectedQuayCranes="1,␣2,␣3␣" selectedStorageBlocks="1,␣2,␣3,␣4␣" />
55      </vessel>
56      <vessel berth="3" appearanceTime="5000" estimatedEndOfLoading="41360.0"
            departureTime="43360.0" vesselType="M␣−␣1600" numberOfQuayCranes="5" />
57    </vesselList>
58    <technical>
59      <quayCranes>
60        <hoistingSpeed>2.1666666666666665</hoistingSpeed>
61        <trolleySpeed>3.6666666666666665</trolleySpeed>
62      </quayCranes>
63      <chassis>
64        <speed>5.0</speed>
65      </chassis>
66      <gantryCranes>
67        <speed>2.0</speed>
68        <hoistingSpeed>0.75</hoistingSpeed>
69        <trolleySpeed>1.6666666666666667</trolleySpeed>
70      </gantryCranes>
71      <liftingTrolleys>
72        <speed>1.0</speed>
73        <hoistingSpeed>0.8333333333333334</hoistingSpeed>
74        <spreaderRotationTime>5.0</spreaderRotationTime>
75      </liftingTrolleys>
76      <flatTrolley>
77        <speed>5.0</speed>
78      </flatTrolley>
79      <general>
80        <containerPickUpTime>5.0</containerPickUpTime>
81        <containerDropOffTime>5.0</containerDropOffTime>
82      </general>
83    </technical>
84    <stochastic>
85      <containerSeed>100</containerSeed>
86      <orderAllocationSeed>100</orderAllocationSeed>
87      <gantryDelaySeed>100</gantryDelaySeed>
88      <automaticSeed>false</automaticSeed>
89      <vesselDelay>
90        <probability>0.0</probability>
91        <extraTime>125</extraTime>
92      </vesselDelay>
93      <truckDelay>
94        <probability>0.0</probability>
95        <extraTime>125</extraTime>
96      </truckDelay>
97    </stochastic>
98  </parameters>
```

A.3. Inhalt der beigefügten DVD

- das in Desmo-J erstellte Modell
  - in Form einer ausführbaren .jar Datei
  - als Sourcecode
- die Modell-Layout-Datei
- die 3D-Komponenten
- Deflault-Parameter-Dateien
  - Extra.ani
  - DefaultParameter.xml
- Auswertungsdateien der in Kapitel 7 diskutierten Szenarien
- Präsentationsfilme des Herstellers zum Horizontalkrankonzept
- Quellen

A.4. Anschrift des Autors und der Betreuer

Autor: **Nikolaus Bornhöft**

Harburger Chaussee 63
20539 Hamburg

Email: Klaus@neuschoe.de

Estbereuer: **Prof. Dr.-Ing. Bernd Page**

Arbeitsbereich „Angewandet und sozilorientierte Informatik“
des Departments Informatik der Universität Hamburg

Vogt-Kölln-Straße 30
D-22527 Hamburg

Zweitbetreuer: **Dr.-Ing. Holger Schütt**

Institut für Seeverkehrswirtschaft und Logistik (ISL)

Universitätsallee 11 - 13
28359 Bremen

Glossar

AC3D Programm der Firma Invis zum Erstellen von 3D-Objekten

AGV Automatisch geführtes Fahrzeug, engl. Automated Guided Vehicle; Transportchassis auf einem Containerterminal, das fahrerlos Containertransporte durchführt

Berth engl., Liegeplatz; Bereich des Kais, den ein vor Anker liegendes Schiff belegt

Berthing-Area Kaibereich eines Containerterminals, dieser umfasst die Liegeplätze der Containerschiffe sowie die Kaikrane

Box ein Container, von dessen Größe abstrahiert wurde

Chassis engl., Fahrgestell; hier Transportfahrzeug für ebenerdigen Containertransport innerhalb des Terminals; verbindet im Horizontalkrankonzept Low-Frame-Bridge und Stacking-Area

Container-Slot auch Slot, dreidimensionale Stellplatzangabe in einem Containerblock, setzt sich zusammen aus der Angabe des Containerstapels (Stack) und der Ebene des Containers (Tier)

Container-Vessel	engl.; Containerschiff
Containerblock	auch Block; Einheit des internen Containerzwischenlagers im Containerterminal. Ein Containerblock besteht aus mehreren Reihen von Containern, die mehrere Ebenen hoch gestapelt sind und von einem Portalkran überspannt werden
Containerbrücke	Kran zum be- und entladen von Containerschiffen
CT	Abkürzung für Containerterminal
Dual-Lift	Verwendung einer Containerbrücke mit einer Katze, mit zwei unabhängig von einander beweglichen Spreadern
Feeder-Vessel	engl.; auch: Feeder, Feederschiff; kleines Containerschiff, das für das Containerterminal als Zubringer dient und auch kleinere und Binnenterminals anfahren kann
Flat-Trolley	Teil des Horizontalkrankonzeptes; Schlitten, der sich über eine Horizontalspur der Low-Frame-Bridge bewegt und Container transportiert
Gantry-Crane	engl.,Portalkran: Kran, der in einen Containerblock der Stacking-Area Container ein- und auslagert.
Gate	engl., Tor: Anlage zum be- und entladen der landseitig angebundendne Transportmittel LKW und Eisenbahn
Kaikran	engl. Quay Crane; Hafenkran zum be- und entladen von Schiffen, hier synonym zu Containerbrücke verwendet

Katze siehe Laufkatze

Laschen Entfernen der Transportsicherungen wie Twist-Locks von eingeschifften Containern

Laschplattform Arbeitsebene auf Zwei-Katz-Brücken, sie bildet den Übergabepuffer zwischen den beiten Laufkatzen und dient gleichzeitig als Arbeitsbereich für das Laschen

Laufkatze engl. Trolley; Krankomponente, die sich entlang einer horizontalen Achse bewegt und über die der Absetzer des Krans bewegt wird

Lifting-Trolley Teil des Horizontalkrankonzeptes; kleiner Kran, der sich auf einer Spur der Low-Frame-Bridge bewegt und dort Container von Flat-Trolleys auf Chassis bzw. umgekehrt umlädt

Low-Frame-Bridge Teil des Horizontalkrankonzeptes; unterhalb des landseitigen Übergabebereichs der Containerbrücken gelegener ein- oder mehrspuriger Stahlrahmen, auf dem Flat- und Lifting-Trolleys arbeiten

Portalkran siehe Gantry-Crane

Prestacking engl., Vorstau; Einlagern von Container in einem bestimmten Bereich der Stacking-Area, damit diese später für ihren Weitertransport entweder schneller zugreifbar sind oder nur noch kürzere Wege innerhalb des Terminals zurücklegen müssen

Reach-Stacker Flurförderzeug, das mit Teleskoparm und Containerheberahmen Container innerhalb des Terminals hebt und transportiert

Restacking	engl., Umlagern; Umlagern eines Containers, entweder um an darunter liegende Container zu gelangen, oder im Rahmen des Prestackings (siehe dort)
Spreader	engl., Heberahmen, der als Teil eines Krans zum Aufnehmen von Containern dient
Stack	engl., Stapel; innerhalb eines Containerblocks: ein Container breiter und ein Container tiefer Stapel von mehreren Containern; auch der Containerblock selbst (siehe dort)
Stacking-Area	der Bereich eines Containerterminals in dem Container bis zu ihrem Weitertransport zwischengelagert werden
Straddle-Carrier	engl., Portalhubwagen; Containerterminalbetriebsmittel, das Container sowohl heben, als auch transportieren kann
Terminal-Chassis	siehe Chassis
TEU	engl. twenty feet equivalent unit, die 20-Fuss-Einheit; Kapazitätsangabe für Standardcontainer
Tier	engl. Containerlage; Ebene, in der ein Container in einem Block liegt
Twistlock	engl., Riegelbolzen; dient zur festen Arretierung zweier Container auf einander zur Sicherung
Van-Carrier	siehe Straddle-Carrier
Vorstau	Konzentration von Containern, die auf ein Schiff geladen werden sollen in einem bestimmten Bereich eines Containerterminals oder eines Containerblocks

XML Extended Markup Language, Auszeichnungssprache zur semiformalen Beschreibung von Daten

Yard engl., siehe Stacking-Area

ZPMC Kranhersteller aus China, entwickelte das Horizontalkrankonzept

Literaturverzeichnis

[Abe] ABENDBLATT.DE: Ankunft der Giganten: Fünf Riesenbrücken für Eurogate. 1.Juni 2007. http://www.abendblatt.de/hamburg/article470238/Ankunft-der-Giganten-Fuenf-Riesenbruecken-fuer-Eurogate.html, Abruf: 29.10.09, 15:00 Uhr

[ASF04] APACHE SOFTWARE FOUNDATION, The: *Apache License.* http://www.apache.org/licenses/LICENSE-2.0. Version: 2004, Abruf: 5.12.09. – Version 2.0

[Bet] BETEILIGUNGSREPORT.DE: *Feederschiff.* http://www.beteiligungsreport.de/fileadmin/user_upload/Bilder/Galerie//Schiffstypen/02Feederschiff.jpg, Abruf: 28.10.09, 16:10 Uhr

[Bil] BILDUNG, Bundeszentrale für p.: *Entwicklung des grenzüberschreitenden Warenhandels.* Internet. http://www1.bpb.de/wissen/CCDNXF,0,Entwicklung_des_grenz%FCberschreitenden_Warenhandels.html, Abruf: 6.12.09, 16:00 Uhr

[BK08] BORNHÖFT, Nikolaus A. ; KLÜCKMANN, Felix: *Softwarevergleich von DESMO-J mit Plant Simulation anhand eines Schleusenbeispiels.* Projektausarbeitung, 2008

[Con] CONTAINERPORTAL.DE: *Die verschiedenen Containertypen.* http://www.containerportal.de/content/containertypen.htm, Abruf: 30.10.09, 16:00 Uhr

[des] *DESMO-J - Homepage.* http://desmoj.sourceforge.net/home.html, Abruf: 4.12.09

[ECT] ECT: *Europe Container Terminals - ECT Delta Terminal.* http://www.ect.nl/frames.asp?currentItemCode=2_1&newwindow=0&subNav=false&currentServiceID=null&currentPageID=22, Abruf: 8.10.09, 16:30 Uhr

[Fle08] Flecks, Joachim: Wettbewerbsfaktor Terminalkapazitäten: Neue Prognosen für die Containerschifffahrt. In: *HypoVereinsbank, Global Shipping Division* Februar (2008)

[GDV] GDV: *Containerhandbuch - Fachinformationen der deutschen Transportversicherer.* http://www.containerhandbuch.de/chb/stra/index.html?/chb/stra/stra_01_03_01_02.html, Abruf: 27.10.09, 15:00 Uhr

[Gie04] Giemsch, Peer: Containerumlade- und Stapelprobleme Universität Karlsruhe, Institut für Anwendungen des Operations Research, 2004

[Goo] *Google Earth.* Internet. http://earth.google.com/intl/de/, Abruf: 6.12.09, 20:00 Uhr

[Gota] Gottwald: *Automated Guided Vehicles AGV – die Zukunft hat bereits begonnen.* http://www.gottwald.com/gottwald/site/gottwald/de/products/agv.html, Abruf: 28.9.09, 16:00 Uhr

[Gotb] Gottwald, Port Technology G.: *Automated Container Transport - Proven Technology from Gottwald.* Online-Prospekt. http://www.gottwald.com/gottwald/export/gottwaldsite/de/news/pdf/AGV_Prospekt_uk.pdf, Abruf: 3.12.09

[Har02] Harold, Elliotte R.: *Processing XML with Java - A Guide to SAX, DOM, JDOM, JAXP and TrAX.* Addison-Wesley Professional, 2002. – 1120 S.

[HHLa] HHLA: *Gottwald Batteriy AGV.* http://www.hhla.de/uploads/tx_otoldnewgallery/Gottwald_Batterie_AGV_02.jpg, Abruf: 12.10.09, 15:00 Uhr

[HHLb] HHLA: *Hamburger Hafen und Logistik AG: Technische Daten, Altenwerder.* http://www.hhla.de/Altenwerder-CTA.64.0.html, Abruf: 28.9.09, 15:10 Uhr

[HHLc] HHLA: *Hamburger Hafen und Logistik AG: Technische Daten, Burchardkai.* http://www.hhla-cta.de/Burchardkai-CTB.63.0.html, Abruf: 28.9.09, 15:00 Uhr

[HS00] Hartmann, Sönke ; Schütt, Holger: Simulation in Planung, Realisierung und Betrieb am Beispiel des Container-Terminal Altenwerder. In: Möller, D.P.F. (Hrsg.): *Simulationstechnik, 14. Symposium.* SCS Europe BVBA, Ghent, 2000

[Inf] *Double Rail-mounted Gantry Crane.* http://www.inform-ac.com/de/or-hafen.htm, Abruf: 30.10.09, 18:30 Uhr. – Inform GmbH - Institut für Operations Research und Management

[ISLa] ISL: *3D Realtime Animation Player (Golem 3D).* – Version 1.7.0.15

[ISLb] ISL: *Templateserstellung mit dem ISL Layouteditor.* – Autor unbekannt

[ISLc] ISL: *Verdigo - Scusy Layout Editor.* – Version 1.0.2.3

[ISL08] ISL: *Golem 3D - Dokumentation der Datenschnittstelle.* 1 2008

[Kon] Konecranes: *Straddle Carrier.* http://www.konecranes.de/produkte/straddlecarrier/, Abruf: 30.9.09, 17:30 Uhr

[ME07] McLaughlin, Brett D. ; Edelson, Justin: *Java & XML.* O'Reilley, Beijing, Cambridge, Farnham, Köln, Paris, Sebastopol, Taipai, Tokyo, 2007

[Mit] Mitic, Katja: Containerverkehr am Burchardkai wird von 2010 an neu geregelt. In: *Welt online* 22. August 2008. http://www.welt.de/welt_print/article2338568/Containerverkehr-am-Burchardkai-wird-von-2010-an-neu-geregelt.html, Abruf: 4.11.09, 18:10 Uhr

[Nau] Nauticexpo: *Frachtschiff : Post-Panamax Containerschiff DAEWOO SHIPBUILDING 8400 TEU.* http://www.nauticexpo.de/prod/daewoo-shipbuilding/frachtschiff-post-panamax-containerschiff-30890-194193.html

[NTB] *Northsea Terminal Bremerhaven.* http://www.ntb.eu, Abruf: 28.9.09, 14:10 Uhr

[Pag91] PAGE, Bernd: *Diskrete Simulation - Eine Einführung mit Modula-2*. 1. Springer-Verlag, Berlin, Heidelberg, New York, London, Tokyo, Hong Kong, Barcelona, Budapest, 1991. – 502 S.

[PK05] PAGE, Bernd ; KREUTZER, Wolfgang: *The Java Simulation Handbook – Simulating Discrete Event Systems with UML and Java*. Shaker Verlag, Hamburg, 2005

[Sch89] SCHOTT, Rainer: *Verkehrswissenschaftliche Studien aus dem Institut für Verkehrswissenschaft der Universität Hamburg*. Bd. Band 39: *Stauplanung für Containerschiffe*. Vandenhoeck & Ruprecht, 1989

[Sch04] SCHMIDT, Axel de: Durchblick im Containerdschungel. In: *Logistik Heute* (2004), Nr. 8, S. 26–27

[Sch08] SCHIECK, Arno: *Internationale Logistik: Objekte, Prozesse und Infrastrukturen*. Oldenbourg Wissenschaftsverlag, 2008. – 467 S.

[Siea] SIEMENS: *Automatisierungs- und Antriebstechnik für mehr Effizienz und Produktivität*. Internet. `http://www.automation.siemens.com/_de/portal/index.htm`, Abruf: 16.11.09, 15:00 Uhr

[Sieb] SIEMENS: *Referenz - Rail Mounted Gantry (RMG) Krane: Kran Service Rheinberg, Demag; HHLA Burchardkai, Deutschland*. Internet. `http://www.automation.siemens.com/mc/cranes/de/5a6ca5fc-e93a-11d6-86e5-080006278927/index.aspx?object=a1e0e28c-59e1-4d3d-87e8-8abe42bc817f`, Abruf: 16.11.09, 16:41 Uhr

[Siec] SIEMENS: *Referenz Containerbrücke: ZPMC; HHLA Container Terminal Altenwerder, Deutschland*. Internet. `http://www.automation.siemens.com/mc/cranes/de/fb30e335-32b7-4a47-b9c4-f6cb71ea2834/index.aspx?object=9c463cf5-9d09-4b90-8ab1-f575098738ad`, Abruf: 16.11.09, 16:00 Uhr

[Sied] SIEMENS, PLM S.: *Plant Simulation*. Internetseite,

[Tec] TECHNOMATICS: *Plant Simulation Benutzerhandbuch*. – Version 8.1

[Ull09] Ullenboom, Christian: *Java ist auch eine Insel*. 8. Galileo Computing, 2009. – 1.475 S. http://openbook.galileocomputing.de/javainsel8/

[Wika] Wikipedia: *Containerschiff*. http://de.wikipedia.org/wiki/Containerschiff, Abruf: 28.10.09, 13:30 Uhr

[Wikb] Wikipedia: *Feederschiff*. http://de.wikipedia.org/wiki/Feederschiff, Abruf: 28.10.09, 12:30 Uhr

[ZPMa] *ZPMC-Homepage*. http://www.zpmc.com, Abruf: 25.9.09, 14:40 Uhr

[ZPMb] ZPMC: *The Environmentally Friendly Automated Container Yard Stacking System*. Präsentationsfilm,

[ZPMc] ZPMC: *The Environmentally Friendly Automated Container Yard Stacking System - Introduction of Changxing Island Experimental Demostration Area*. Präsentationsfilm,

Abbildungsverzeichnis

„Ich versichere, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt habe und mich keiner anderen als der im beigefügten Verzeichnis angegebenen Hilfsmittel bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Alle Quellen, die aus dem World Wide Web entnommen oder in einer sonstigen digitalen Form verwendet wurden, sind der Arbeit beigefügt."

Nikolaus Alexander Bornhöft

Hamburg den 7. Dezember 2009.